Столлмен: Самое значимое интервью

Плюс: Jolicloud 1.2, Gnome 3 Live, Trisquel + еще много чего!

LINUX FORMAT

Главное в мире Linux

Июнь 2011 № 6 (145)

DreamPlugging

NAS, WAP и аудиоплейер величиной с электровилку с. 60

В ЭТОМ НОМЕРЕ

Советы по Linux

» Подробнейшие учебники



Ваш дистрибутив без излишеств

» ArchLinux затянут туго

Нырнем в Jolicloud

» Что нового в дистрибутиве для нетбуков с быстрым Интернет



Спасибо вам за музыку

» Тест четверки лучших плейеров

Битва накопителей

» Qnap 219P+ Linux NAS бросил вызов Synology DS211+



GNOME 3.0 с нами!

Все, что следует знать про первый главный релиз за 8 лет

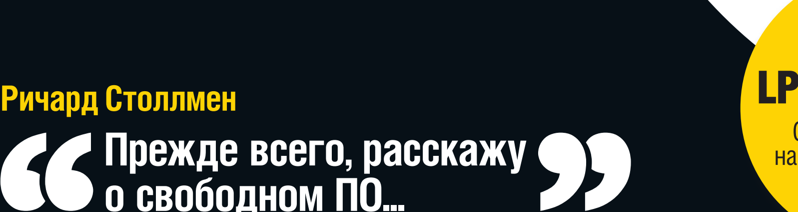


» Освоим Gnome Shell

» Поработаем в GTK+ 3

» Портируем настройки

» Освежим рабочий стол



Ричард Столлмен

“Прежде всего, расскажу о свободном ПО...”

Мы преломили хлеб с крестным отцом GNU с. 32



Плюс!

LPI: где и как

Сдача экзаменов на Linux-сертификат в России с. 42

4 607130 825167 11006

Ключевые навыки

Потолкуем с Gmail

» Родные средства Linux управляют сетевой почтой

Скрипты

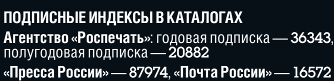
KDE автоматически

» Покорите рабочий стол, не прикладая мыши

Сисадмину

Установка через LAN

» Обновите машины в своей сети удаленно


ПОДПИСНЫЕ ИНДЕКСЫ В КАТАЛОГАХ
Агентство «Роспечать»: годовая подписка — 36343, полугодовая подписка — 20882
«Пресса России» — 87974, «Почта России» — 16572





Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

Кто мы

В этом выпуске мы поговорили с гуру свободного ПО Ричардом Столлменом. Мы спросили нашу команду: какой бы вы задали вопрос, беря интервью у РМС?



Майк Сондерс
А, вы тот самый парень насчет открытого кода, который изобрел Linux? Просто чтобы засечь выражение его лица.



Эндрю Грегори
Почему FAQ про именование GNU/Linux на вашем сайте длиннее, чем Конституция США?



Эфраин Эрнандес-Мендоса
Правда ли, что вы на самом деле потерявшийся в детстве брат-близнец Алана Кокса?



Нейл Ботвик
Столлман – гений и знает все; вот я и спрошу его, куда же я засунул ключи от своей машины.



Маянк Шарма
Я бы спросил его о чем-нибудь, используя регулярные выражения. Эй, Ричард, /:[..*...



Д-р Крис Браун
Ваш IPad черный или белый? И какой у вас высший балл в игре *Angry Birds*?



Энди Ченнел
А можно, я сделаю death-metal-версию «Песни свободного ПО»? И вставлю шум и выкрики?



Валентин Синицын
Не беспокоит ли вас, что у меня сейчас в кармане карта Oyster и мобильник? Вдруг это заразно?



Ник Вейч
Величайший, абсолютно насущнейший вопрос всех времен: *Wine* – это все-таки эмулятор или нет?



Сюзан Линтон
Неужто вы и правда такой умный, что можете использовать 0, 1 *и* 2 при двоичном подсчете?



Шашанк Шарма
У вас все еще тот PDP-10, на котором вы разработали *Emacs*? А работает ли на нем *Crysis*?



Боб Мосс
Надо ли вам запускать **su**, или это происходит автоматически, просто потому, что вы РМС?



Слушайте команду

В этом номере опубликовано сразу два учебника о работе в командной строке. Один из них – продолжение учебного курса Майка Сондерса о подготовке к сертификации LPI, а другой откроет новый цикл учебников нашего постоянного автора Тихона Тарнавского. Любопытно, что сам подход к обучению у авторов отличается «с точностью до наоборот» – мы заинтересовались возможностью осветить предмет с разных сторон, скрестив на нем прожектора с противоположных краев Европы.

Разговор о командной строке и истории UNIX наводит на мысли об одной из главных причин успеха этой системы, а именно – оптимальном сочетании развитого языка командного процессора (shell) и богатых возможностей набора утилит GNU, позволяющих работать с текстовыми строками.

А если кому-то покажется, что работа в командной строке трудна и неудобна, взгляните на фотографию компьютера DEC PDP-7, для которого и был написан первый UNIX. На ней хорошо виден исторически первый /dev/tty0. Сегодня многие уже не помнят, что это устройство называется телетайп, откуда и пошло сокращение /dev/TeleTYpe. Другого интерфейса в те времена не было, но это не помешало создать великую систему, которой удалось не устареть за 40 лет – для программного обеспечения целая эпоха. Так что, нажимая Ctrl+Alt+F1, вспомните об истоках.

И – возвращаясь к истокам: мы начинаем публикацию серии статей Алексея Федорчука, посвященных истории Linux, которому осенью исполняется 20 лет.

Кирилл Степанов
Главный редактор
» info@linuxformat.ru

Как с нами связаться

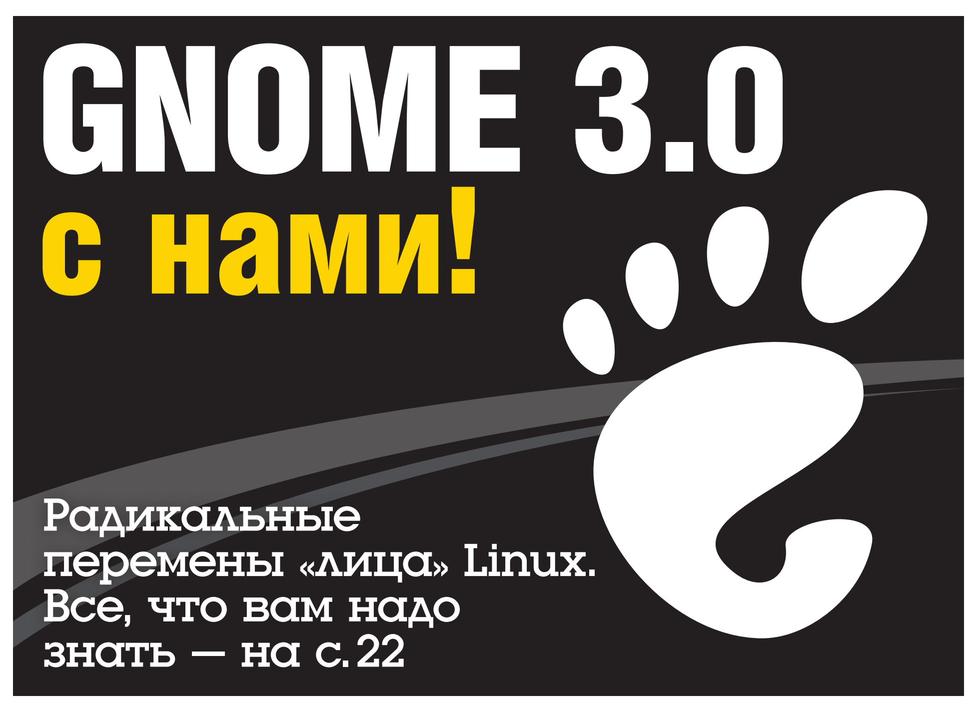
Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Вопросы распространения: sales@linuxformat.ru
Web-сайт: www.linuxformat.ru
› Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15
› Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112



Содержание

Гномов больше, чем в садике на детской площадке.

Обзоры



GNOME 3.0 с нами!

Радикальные перемены «лица» Linux. Все, что вам надо знать – на с. 22



ТРЕНИНГИ LINUX

Повысите квалификацию с Майком Сондерсом – повысят и вас с. 64



Приведите свою жизнь в порядок с Джоно Бэконом и Tomboy с. 38

Люди говорят



«Идея свободного ПО в том, чтобы избежать несправедливости и положить ей конец»

Ричард Столлмен рассказал нам про свободу с. 32

На вашем бесплатном DVD

Gnome 3, Jolicloud и другие

LXF DVD145

» **Gnome 3:** готовьтесь увидеть нечто абсолютно новое
» **Arch Linux:** как получить то, что надо лично вам
» **Jolicloud:** Летим в облака!

ПЛЮС: Trisquel, игры, PDFы, подкасты... с. 100

Ищите в этом номере...

Пропустили номер?

Узнайте на с. 107, как получить его прямо сейчас!



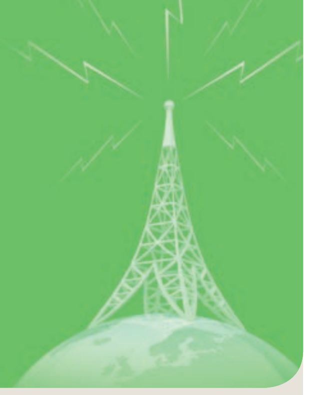
Постоянные рубрики

Учебники



LINUX FORMAT

Новости



ГЛАВНОЕ Сделка по Novell закрыта » SUSE будет самодостаточным » Минюст США за свободное ПО » Война Google и Oracle » Патенты — все-таки зло?

ПАТЕНТЫ НА ПО

Обострение патентной проблемы



» Рубрику готовил **ТИМУР МУБАРАКШИН**

Novell теперь официально продана. Но ход сделки показал, что патенты в индустрии ИТ становятся угрозой.

Novell продана. Теперь уже официально. В конце апреля сего года сделка была закрыта. Продажа такой компании, как Novell, имеет очень большое значение для всей индустрии ИТ, и, конечно, повлияет на мир СПО. Вопрос в том, как? Мнения экспертов расходятся. Одни считают, что это пойдет на пользу Novell, другие утверждают, что эта сделка стала началом конца Linux. Пока рано делать какие-либо выводы, однако уже сейчас понятно, что старой Novell уже не будет. Эта сделка показала, что мир СПО пока еще очень сильно уязвим, что существует большая «патентная проблема», и необходимо уже сейчас выработать эффективные механизмы защиты от «патентных троллей». Защитить Open Source и Linux может только широкая коалиция всех неравнодушных людей и организаций. Оказалось, что такие есть не только среди программистов, но и в правительствах США и Германии. Суть такова: Министерство юстиции США одобрило сделку по продаже Novell при условии соблюдения ряда условий, а именно: патенты будут приобретены консорциумом CPTN, но не смогут быть использованы против экосистемы Linux и GPL-проектов. Компания Microsoft, после получения своей 25 % доли, вернет патенты компании Attachmate, при этом последняя предоставит Microsoft лицензию на все переданные патенты. Доли Apple, EMC и Oracle составят 25 %, при этом для EMC (контролирующая *VMware*) введен запрет на получение 31 патента, имеющих отношение к виртуализации. Передаваемая CPTN интеллектуальная собственность не связана с правами на Unix, все имущественные права на Unix остаются в руках бизнес-подразделения Novell. Все необходимые для развития проектов SUSE и openSUSE технологии, права и интеллектуальная собственность остаются в руках Attachmate.

Бизнес-подразделение SUSE будет самодостаточным и объединит все ресурсы, интеллектуальную собственность, персонал и инфраструктуру для формирования и распространения продукта. С организационной точки зрения мало что изменится. Финансирование и развитие проекта openSUSE будет продолжено. Attachmate также продолжит участие в разработке Linux-ядра и поддержку других начинаний Novell, связанных с продвижением решений на базе открытого кода, среди которых KDE, GNOME, *X.org*, MeeGo, Linux-ядро, Mono и *LibreOffice*.

> «Защитит Open Source только широкая коалиция неравнодушных.»

Такие, мягко говоря, невыгодные для Microsoft и ее партнеров условия сделки стали возможными благодаря усилиям организаций Open Network Invention (ONI), Open Source Invention (OSI) и Free Software Foundation. В декабре прошлого года ONI (Open Network Invention) обращалась с письмом в антимонопольный комитет Германии, а в январе этого года фонд Free Software Foundation попросил минюст США вмешаться в процесс сделки, поскольку подобные не публичные сделки могут быть использованы для «сокрытия нечестивых намерений». Минюст, работая в тесном сотрудничестве с антимонопольным комитетом Германии, согласился с точкой зрения OSI и FSF и потребовал капитальных изменений. OSI – это компания, владеющая патентами на ПО экосистемы GNU/Linux и предоставляющая свободные от отчислений лицензии на право использования этих патентов. Членами этой организации, среди прочих, являются: IBM, Novell. Philips, Red Hat, Sony и Google.

› **openSUSE выйдет из-под руководства Novell, став самостоятельным проектом со штаб-квартирой в Германии.**

Интерес к правовой защите открытых технологий возрастает. Так, организация Open Invention Network (OIN) объявила о значительном росте интереса компаний к инициативе по защите Linux от патентных претензий и обмене связанной с Linux интеллектуальной собственностью. Утверждается, что за первый квартал 2011 года число участников организации увели-

чилось на 28 % – к OIN присоединились 74 компании, использующие в своей деятельности Linux и желающие снизить риски от предъявления возможных патентных претензий. Общее число вовлеченных в работу OIN компаний достигло 334. Примечательно, что среди 74 новых участников OIN – не только небольшие компании, курирующие развитие свободных проектов и оказывающие услуги поддержки, но и таких гиганты индустрии, как Hewlett-Packard, Fujitsu, Juniper Networks, Rackspace Hosting, Symantec и Facebook. Подключение подобных компаний к инициативе существенно увеличивает возможности OIN по защите Linux от патентных агрессоров и подчеркивает состоятельность идеи о возможности обмена связанными с интеллектуальной собственностью ресурсами с целью созидания и обеспечения совместного развития перспективных технологий. Защита у мира открытого ПО уже есть; главное – использовать ее с умом.

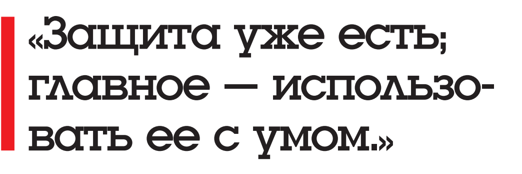
В качестве основного стимула, подтолкнувшего к вступлению в OIN большого числа различных компаний в начале года, стало заявление о поглощении компании Novell и продаже 882 патентов консорциуму CPTN Holdings LLC, подконтрольному Microsoft, Apple, EMC и Oracle. По мнению правозащитного ресурса Groklaw, вступив в организацию OIN до завершения сделки, компании могут обезопасить себя от возможных патентных претензий в будущем. Уже заключенное соглашение об обмене интеллектуальной собственностью носит бессрочный характер и будет действовать для членов OIN даже после перехода интеллектуальной собственности Novell другому владельцу, так как Novell входит в состав OIN. Действенность соглашения OIN также была подтверждена решением Министерства юстиции США, потребовавшим учесть интересы OIN в условиях сделки о продаже патентов Novell.

В связи с этими событиями интересна другая «патентная» новость. Компания Barnes & Noble ответила на предъявленный компанией Microsoft иск, в котором указывается на нарушение патентов Microsoft при использовании платформы Android в электронных книгах Nook.

По словам представителей Barnes & Noble, у Microsoft есть план дискредитации Android в глазах производителей мобильных устройств и потребителей, путем установки «непомерных лицензионных сборов и абсурдных лицензионных ограничений». Все это направлено на доминирование собственной мобильной платформы Microsoft. Сделка с Nokia является развитием этой нечестной схемы, считают в Barnes & Noble. Причем размер лицензионного сбора, который Microsoft пытается получить с производителей устройств на базе платформы Android, превышает стоимость мобильной версии Windows 7.

В Barnes & Noble уверены, что указанные в жалобе Microsoft патенты на самом деле являются «незначительными, не нарушенными и недействительными» и служат лишь средством достижения корыстных целей Microsoft. Указанные патенты нисколько не связаны с функциями, реализованными в электронной книге Nook от Barnes & Noble и в других устройствах компании, а также в Google Android.

Google и Oracle продолжают патентную войну. Компания Oracle получила некоторое преимущество в разбирательстве, в котором компания Google обвиняется в нарушении прав на связанную с Java интеллектуальную собственность. Окружной судья согласился с интерпретированием Oracle четырех из пяти патентных технических терминов. Данная процедура также называется «искусством перевода жаргона патентных притязаний на разговорный английский». Это означает, что технические термины и их семантическая интерпретация разъяснены и разграничены для оперирования ими на следующих заседаниях.


«Защита уже есть; главное — использовать ее с умом.»

Толкование патентных формул крайне важно, так как на его основании суд должен принять точку зрения одной из сторон.

Слушание дела ожидалось в ноябре, однако, согласно последним сообщениям, обе компании, Oracle и Google, по требованию судьи, подали предложения о сокращении числа исков о возмещении ущерба в связанном с Java патентном деле, что может ускорить завершение дела. Oracle предложила сократить число исков в итоге до 35 ко 2 сентября, а затем, к 17 октября и дате предварительного досудебного совещания, до количества, «подлежащего рассмотрению в суде». Также Oracle хочет, чтобы Google до 8 июля снизила количество утверждений о ссылках на известный уровень техники до четырех на патент, и сократила количество указаний на юридически несостоятельные основания до трех на патент ко 2 сентября.

В свою очередь, Google значительно сократит количество утверждений о ссылках на известный уровень техники. Поданные заявления также касаются текущей повторной проверки патентов Oracle патентным ведомством США, инициатором которой была Google. Oracle протестует против замораживания или отложения дела в связи с этими перепроверками, утверждая, что данные перепроверки не принесут ощутимых плюсов ни суду, ни сторонам, и только лишь серьезно нарушат расписание заседаний, утвержденное судом. Согласно Oracle, скорей всего данные перепроверки будут закончены не раньше конца 2013 года, но Google ожидает, что патентное ведомство выдаст некоторое количество информации по этому вопросу в течение ближайших нескольких месяцев.


› Главное здание Министерства юстиции США — здесь встали на защиту Open Source от патентных троллей.

Разборки, происходящие вокруг патентов на интеллектуальную собственность высокотехнологичных компаний, уже получили адекватную оценку у чиновников США – Федеральная торговая комиссия опубликовала отчет, в котором констатируются важные проблемы патентной системы США, в частности, отрицательно влияющие на индустрию разработки ПО. В отчете также предлагаются некоторые реформы, которые потенциально могут снизить ущерб от некоторых из этих проблем.

Многие люди из ИТ негативно относятся к идее патентования программного обеспечения: например, генеральный директор Red Hat Джим Уайтхёрст [**Jim Whitehurst**] в своем недавнем интервью заявил, что ненавидит так называемых «патентных троллей», но иногда приходится соглашаться с их требованиями ради блага компании. Red Hat сейчас одна из самых успешных компаний OpenSource, которая смогла заработать на открытых технологиях почти миллиард долларов и продолжает показывать устойчивый рост. Однако проблемы с патентными троллями не обошли стороной и этого гиганта, и продолжают, по словам Уайтхёрста, представлять для них серьезную проблему.

Надо что-то делать

Проблема патентных споров требует возможно более скорого решения, иначе могут пострадать все – и традиционные «проприетарные» компании, и такие сторонники открытого ПО, как Google или Red Hat. Патенты из средства защиты превратились в инструмент запугивания и шантажа; особенно этим «грешат» Microsoft и Oracle.

СВОБОДНОЕ «ЖЕЛЕЗО»

Rally Fighter и Openfarmtech.org

В мире Open Hardware появились интересные проекты – действительно железные...

Есть на свете открытые автомобили (имеются в виду не кабриолеты!), которые тешат умы и сердца сторонников OpenSource. Однако привлекательность дизайна не относится к их сильным сторонам, поэтому особой популярностью они не пользовались. В основном эти машины строились на шасси раллийных багги и выглядели, мягко говоря, слишком утилитарно. Ситуация может измениться с появлением Rally Fighter – полностью открытого автомобиля. Разрабатываемая сообществом под руководством небольшой автомобильной компании Local Motors (Бостон, штат Массачусетс, США; основана в 2007 г.), гоночная раллийная машина предназначена для езды преимущественно по песчаным трассам, однако, по заверениям разработчиков, может использоваться и на дорогах общего пользования. Автомобиль обладает эффектным дизайном в стиле “muscle car” и пока продается лишь в виде готового к сборке набора за $59000, что, конечно, совсем недешево. В будущем планируется наладить серийный выпуск; при этом цена будет ниже, чем у комплекта для сборки.

В компании Local Motors считают, что автопроизводители на самом деле не слушают своих клиентов, поэтому они решили сделать все по-другому. Джей Роджерс [Jay Rodgers], один из соучредителей компании, уверен, что создание сообщества, совместная разработка с этим сообществом, применение голосования по вопросам дизайна и разработки автомобиля обеспечивают жизнеспособный вариант и являются залогом успеха.

Вместо того, чтобы разрабатывать машину целиком, в компании решили разделить создание различных частей автомобиля на самостоятельные проекты. Каждый такой проект имеет ведущего разработчика/дизайнера, который при принятии решений руководствуется результатами голосования, проведенного внутри сообщества. Перед окончательным утверждением какой-либо части машины сообщество еще раз вносит свои улучшения и исправления. При сведении всех проектов вместе каждый участник может предложить изменить все, что он посчитает нужным – в Local Motors считают это свойство Open Source замечательным.

Кроме традиционной модели Open Source, Local Motors также использует модель “crowdsource” – своего рода публичный аутсорсинг, когда работу над задачей/проектом ведет большая группа людей, объединенная Сетью и общими интересами. Часто при этом проводится открытый конкурс на выполнение работ. Если проект Rally Fighter окажется успешным, и другие компании в других отраслях смогут использовать эти модели. Подход сработал в отрасли программного обеспечения, теперь работает в автомобилестроении – почему бы не попробовать эти модели в других отраслях, говорят в Local Motors.



› Это Rally Fighter. Разве не привлекательно выглядит?

> «Этак дойдет и до рабочей модели экономики целого государства!»

Local Motors не считает себя конкурентом традиционных автомобилестроителей. Компания не против использования модели разработки Local Motors современными автопроизводителями и даже вовлечения самой фирмы в управление разработкой будущих моделей автомобилей, так как традиционная разработка машин очень дорогая и требует много времени и усилий. Удачи вам, парни из Local Motors!

Другой интересный открытый проект – Openfarmtech.org, в рамках которого фермеры-энтузиасты сформировали коллекцию распространяемых под свободными лицензиями спецификаций и чертежей, позволяющих из подручных средств собрать своими руками различные машины и устройства – от тракторов и комбайнов до электрогенераторов, станков и даже 3D-принтера. Конечная цель проекта, получившего название Global Village Construction Set (http://openfarmtech.org/wiki/Global_Village_Construction_Set) – формирование полного набора открытых спецификаций, на базе которых можно построить ферму на полном самообеспечении с начальными затратами порядка $10 000.

Open Source становится реально полезным явлением. По открытым спецификациям можно создать любой инструмент, любую технику, и не только для сельского хозяйства. Для многих бедных стран это может стать спасением, да и людям будет чем заняться. Проект Openfarmtech.org стоит трактовать шире, чем просто оборудование для фермы. На самом деле это только начало создания полной технологической цепочки (энергетика, производство материалов, станки и прочие технологии, транспорт, строительство, сельское хозяйство). Этак дойдет и до рабочей модели экономики целого государства!

› Openfarmtech.org — проект по созданию свободной фермы. Но не только!



Не только электроника

Свободное железо – это не только открытые сотовые телефоны и ноутбуки, а знания и технологии, сумма которых позволяет создать полноценную инфраструктуру, самодостаточную и полностью свободную. Долгосрочной целью проекта Openfarmtech.org является создание подобной инфраструктуры. Движемся к полностью свободному миру?

ОБО ВСЕМ ПОНЕМНОГУ

Новости из России

В Москве прошел Russian Open Source Summit 2011.

Мероприятия уровня ROSS проводятся в России нечасто. Саммит прошел при информационной поддержке журнала PC Week Russian Edition. Участники отметили, что российские разработчики слабо вовлечены в общемировые процессы разработки свободного ПО и железа. С принятием Национальной программной платформы, ситуация может измениться к лучшему. Ярким событием ROSS-2011 стали визит и пресс-конференция Марка Шаттлворта, который назвал российский рынок СПО перспективным и привлекательным для разработчиков СПО, похвалил идею создания Национальной программной платформы и пообещал взрастить в России инфраструктуру обслуживания конечных пользователей. Пресс-конференция прошла в День космонавтики – символично для космонавта-линуксоида. В ней также участвовали сотрудники Novell, Red Hat, Canonical (Ubuntu), EnterpriseDB и Alfresco. Участие было бесплатным, при условии предварительной регистрации.

По итогам саммита были сделаны следующие выводы:

» В России огромный потенциал разработки СПО.

» В России мало по-настоящему международных проектов в области СПО.

» В России мало опыта создания сообществ по разработке.

» Русские программисты слабо вовлечены в процесс международной разработки.

» Canonical заинтересована в российском рынке.

В семействе сертифицированных в России дистрибутивов Linux – прибавление: 18 апреля сертификат Федеральной службы по техническому и экспортному контролю (ФСТЭК) получил популярный Альт Линукс СПТ 6.0 – унифицированный дистрибутив для серверов и рабочих станций со встроенными программными средствами защиты информации. Ранее сертификацию ФСТЭК уже проходил ставший практически российским дистрибутив Mandriva.

ФСТЭК – орган исполнительной власти России по реализации государственной политики в области обеспечения защиты информации некриптографическими средствами. Сертификат ФСТЭК позволяет использовать продукт для работы с персональными данными любого типа (в том числе медицинскими), со служебной и конфиденциальной информацией, а также государственной тайной.


СИСТЕМА СЕРТИФИКАЦИИ
СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ

ПО ТРЕБОВАНИЯМ БЕЗОПАСНОСТИ ИНФОРМАЦИИ

СЕРТИФИКАТ СООТВЕТСТВИЯ
№ 2317

[illegible]

ПЕРВЫЙ ЗАМЕСТИТЕЛЬ ДИРЕКТОРА ФСТЭК РОССИИ

» Так выглядит сертификат ФСТЭК.

О чем говорили на ROSS-2011

» СПО в России: текущее состояние, проблемы, достижения и перспективы развития.
» Формирование экосистемы независимых разработчиков.
» Перспективные проекты в области СПО: Готовые системы на базе СПО для различных бизнес-задач и вертикальных рынков.
» Анатомия облака: Готового облака, которое подойдет всем, не существует. Облако – сложная система, охватывающая весь ИТ-стек, от оборудования до прикладных пользовательских приложений. Обсуждение практического опыта построения облачных инфраструктур, а также вопросов стандартизации облачных технологий.
» СПО в корпоративном секторе: С чего начинать миграцию к СПО? Для каких задач открытое ПО является наиболее эффективным?
» СПО в государственном управлении: Обсуждение плана мероприятий по переходу на СПО федеральных органов власти, утвержденного Распоряжением Правительства РФ № 2299-Р.
» Практический опыт использования СПО в российском госсекторе, анализ зарубежного опыта. Вопросы поддержки отечественных разработчиков, ориентированных на СПО.
» Как минимизировать риски при внедрении СПО в госсекторе: Анализ и выработка рекомендаций по предупреждению организационных и технических трудностей на пути миграции госзаказчиков к СПО.
» Стандартизация требований к СПО. Перспективы создания Национальной программной платформы. Обеспечение интероперабельности открытого и «фирменного» ПО.

Новости короткой строкой

» После атаки на сервис PlayStation Network, приведшей к утечке пользовательских данных, инфраструктура Sony была взломана второй раз. Это заставило многих усомниться в безопасности сетевых сервисов Sony.

» 28 апреля 2011 года, как и ожидалось, состоялся релиз Ubuntu 11.04 Natty Narwhal, с коренными изменениями в интерфейсе пользователя.

» Компания Adobe заявила на конференции Google I/O о намерении интегрировать во Flash поддержку открытого формата WebM и связанных с ним медиа-контейнера Matroska, видеокодека VP8 и звукового кодека Vorbis.

» В списке рассылки разработчиков дистрибутива Debian ведется обсуждение возможности создания новой ветки с непрерывным циклом обновления – «rolling release».

» Согласно аналитическому отчету агентства Gartner о развитии рынка операционных систем в 2010 году, самые значительные показатели роста среди серверных систем продемонстрировал Linux. Доминируют среди Linux-систем продукты компании Red Hat, занимающие 58,2 % сегмента Linux-серверов.

» Седьмая международная конференция разработчиков и пользователей свободного программного обеспечения Linux Vacation / Eastern Europe пройдет с 30 июня по 3 июля в Беларуси, недалеко от г. Гродно. Официальные языки конференции – русский, белорусский и английский.

» Nokia планирует сократить 7000 сотрудников, свернуть разработку MeeGo и передать Symbian на аутсорсинг.

» Одна из крупнейших немецких страховых компаний, LVM Versicherungen, объявила о начале миграции 10000 тысяч рабочих станций своих сотрудников в Ubuntu Linux.

» Microsoft приобретает *Skype* за 8,5 миллиардов долларов. На покупку также претендовали Google, Cisco и Facebook, но выдвинутые ими условия оказались менее выгодными.

» Мэтт Циммерман [Matt Zimmerman], бессменный технический директор компании Canonical с момента ее основания, решил попробовать себя на новом поприще. Однако продолжит работу над Ubuntu в качестве волонтера.

В этом выпуске новостей были активно использованы материалы порталов opennet.ru и pcweek.ru, а также свободные источники в Сети.

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
...простой советский юзер.

Куда катится мир?

Все началось с изобретения мониторов широкого формата. Вроде для блага человека – чтобы фильмы комфортней смотреть. Хотя есть и версия, что они просто дешевле в производстве: дисплеев с пропорцией 16:9 из одной «заготовки» нарезается больше. Ну, а нас, пользователей, убедили, что широкоформатный дисплей – это круто. Убедили вполне рыночным методом: сняли с продаж мониторы «квадратные».

Глядь, а для большинства задач, отличных от просмотра фильмов, «длинные» мониторы-то и неудобны: при работе с браузером или текстовым редактором зияют «мертвые боковины». По заветам Ильфа и Петрова, трудности надо преодолевать. И преодолели: создали интерфейсы Gnome Shell и Unity. Каковые, при бесспорной их прогрессивности, фронтирности и прочих достоинствах, большинством «действующих» пользователей Иксов отторгаются. Не беда: надо только убедить пользователей начинающих, что это круто. И ведь убедят – начинающие-то, коими и прирастает могущество Linux'а, имеют шанс ничего иного не увидеть: эти интерфейсы умолчальны в свежевышедшей Ubuntu и грядущей Fedora.

Предвижу отпор прогрессистов: пусть ретроградствующие обскуранты, типа автора этих строк, сидят себе в интерфейсах традиционных. Увы, обновление системы – иногда неизбежность, а не блажь. Но об этом – потом.

P.S. А для задействования «боковин» давно уж придуман *Window Maker*, по сию пору один из самых эстетичных оконных менеджеров. Не послужит ли это стимулом реанимации проекта?

alv@posix.ru

Сегодня мы рассматриваем:

Jolicloud 1.2 с. 11

› Интерфейс напоминает iOS и Android — потому что прицел тот же: на малые сенсорные экраны.

Spotify с. 12



› Чтобы насладиться сладкозвучием *Spotify*, больше не требуется *Wine*.

Наш вердикт: Пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибалльной шкале (0 – низшая оценка, 10 – высшая). Как правило, учитываются функциональность, производительность, удобство использования и цена, а для бесплатных программ – еще и документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту. Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.



Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов в *GCC*, но если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

SpaceChem 1006

Разработчик: Zachtronics Industries
Сайт: www.zachtronicsindustries.com
Цена: $15 (ок. £9,50)

Игровой процесс	10/10
Графика	8/10
Ресурс длительности	10/10
Оправданность цены	10/10

» Затягивающий игровой процесс, сбалансированные трудности и безмерное удовольствие. Чего еще и желать от головоломки?

Рейтинг 10/10

Firefox 4



Несколько засидевшийся на старте *Firefox 4* зато упакован под завязку – и **Ник Вейч** выяснил, что это только начало.

Вкратце

» Самый популярный мультиплатформный браузер с открытым исходным кодом. Аналоги – *Opera*, а также *Chrome* от Google.

Firefox, несомненно, серьезно преобразил рынок браузеров – но теперь, когда прежний лоск пообтерся, не скроется ли матерый лис в норе? Проект уже не производит впечатления флагмана инноваций, как раньше, однако выход версии 4 после долгого сидения в 3.x, похоже, знаменует начало новой амбициозной программы обновления.

Для проверки скорости мы обобщили результаты нескольких web-тестов, включая комплект испытаний JavaScript (SlickSpeed, V8 Benchmark Suite от Google и SunSpider). Быстродействие существенно (в некоторых случаях троекратно) превышает показатели версии 3.6. Отличие от *Chrome* укладывалось в пределы погрешности (кроме как, понятное дело, в собственном тесте Google – V8). Любопытно, что *Firefox* на 5–12 % медленнее работает на Linux, чем на Windows. С *Google Chrome* картина обратная: он «предпочитает» Linux, хотя разница не настолько велика.

> «Важно, что это надежный, безопасный, послушный браузер.»

Впрочем, стандартные тесты редко кто воспринимает всерьез. Скорость по JS годится как показатель, но тем, кто не занимается с утра до вечера тестированием сайтов, разница в глаза не бросится. Она может стать критичной лишь для профессионалов, работающих с web-приложениями. Реальный недостаток *Firefox* в том, что он по-прежнему работает в один поток. То есть при загрузке сайтов с обилием скриптов (или при работе с несколькими сайтами) скорость работы браузера падает с каждой новой вкладкой, независимо от количества ядер процессора.



› Демонстрируя стойкость своей популярности, *Firefox 4* держит марку самого загружаемого пакета в Интернет.

Концептуально *Firefox 4* следует правилу большинства браузеров – «лучше меньше, да лучше». Изменилось расположение верхних панелей инструментов. По умолчанию корешки вкладок располагаются теперь сразу под строкой меню; ниже находится панель навигации, а под ней панель закладок. В итоге верхняя часть нового *Firefox* выглядит несколько перегруженной, однако все это можно отключить.

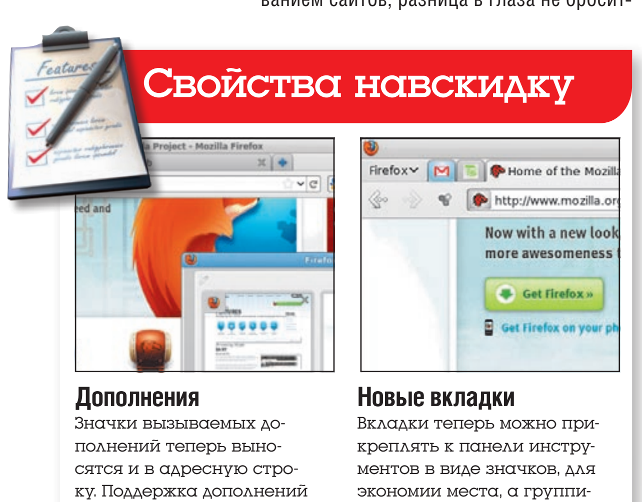
Новые вкладки

Основные изменения касаются вкладок. Как и в *Chrome*, каждую вкладку теперь можно уменьшить до размеров значка и закрепить сбоку окна – это удобно для часто используемых web-приложений. Вкладки можно группировать. Нажав значок группировки в правой части окна, вы получите нечто вроде электронного блокнота, в котором можно сортировать закладки по группам, создавать новые группы (просто очертив прямоугольник!) и переключаться между любыми элементами. Не Бог весть как эффектно, но тем не менее шаг вперед.

Впервые стал поддерживаться выбор заголовка Don't Track – предписания web-серверам не сохранять распознаваемые персональные данные в ходе сеанса. Правда, эту функцию пока признают немногие сайты.

Многочисленные новинки и дополнения появились во всех компонентах, но самое важное – то, что это по-прежнему надежный, расширяемый, безопасный, послушный браузер. Старое знакомство и легионы дополнений будут достаточным стимулом хранить верность этому браузеру – как и одновременный выпуск новой версии для всех платформ. Но в будущем хотелось бы видеть более решительные намерения по направлению развития. LXF

Свойства навскидку



Дополнения

Значки вызываемых дополнений теперь выносятся и в адресную строку. Поддержка дополнений в *Firefox* по-прежнему безупречна.

Новые вкладки

Вкладки теперь можно прикреплять к панели инструментов в виде значков, для экономии места, а группировка упрощает управление ими.

LINUX FORMAT Вердикт

Firefox 4

Разработчик: Mozilla
Сайт: www.mozilla.org
Цена: Бесплатно на условиях GPL

Функциональность	8/10
Быстродействие	8/10
Удобство использования	9/10
Документация	8/10

» Направление верное; но сработает ли избранная стратегия, покажут версии 5, 6 и 7.

Рейтинг 8/10

Jolicloud 1.2





Наконец нашелся дистрибутив, который сделает из вашего нетбука подлинно «сетевую» машину – поверьте **Бобу Моссу**.

Вкратце

» Дистрибутив Linux, рассчитанный на нетбуки и портативные устройства с выходом в Интернет. См. также: Ubuntu NBE и MeeGo.

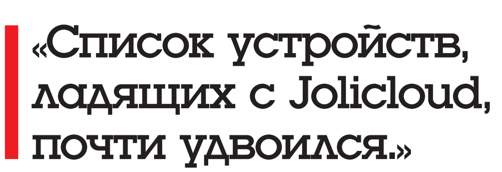
Не прошло и девяти месяцев с тех пор, как Jolicloud 1.0 засветился в нашем обзоре с бодрой оценкой 9/10. Посмотрим, остается ли этот отличный дистрибутив номером один для нетбуков и других устройств с выходом в Интернет.

Обновление в общем оправданно. Jolicloud обзавелся функцией поддержки социальных сетей, позволив отслеживать события у других пользователей – узнавать про новое ПО, интересоваться отзывами о новых пакетах и ощущать себя частью сообщества единомышленников.

Как и в 1.0, можно синхронизировать файлы в своей учетной записи Dropbox на любых устройствах – хотя вариант синхронизации с помощью учетной записи **box.net** убрали, а Ubuntu One так и остается без поддержки.

Список устройств, ладящих с Jolicloud, почти удвоился, так что совместимость с аппаратурой более не проблема.

Ключевая особенность – характер работы с программами. «Склад» приложений


«Список устройств, ладящих с Jolicloud, почти удвоился.»

работает с лету, а все установленные программы и созданные клавиатурные комбинации эта ОС синхронизирует по всем вашим устройствам. Приложения запускаются со значков в главном окне – так же, как на смартфонах с iOS или Android. Этот принцип распространяется на весь интерфейс, обеспечивая простое, интуитивно понятное и быстрое управление.

› **Не удивительно, что интерфейс напоминает iOS и Android — налицо расчет на малые сенсорные экраны.**

Как и прежняя версия, Jolicloud 1.2 отлично подключается к беспроводным сетям. Страницы входа *Chromium* Wi-Fi усовершенствованы, хотя пока не идеальны. Система довольно уверенно чувствует себя на устройствах с ОЗУ объемом 256 МБ, но на 512 МБ работает более гладко.

Тотальная поддержка

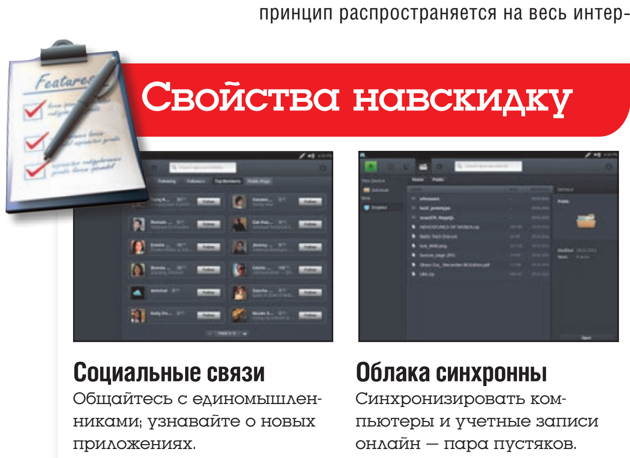
Ваша система будет работать в web-браузере так же, как на самом нетбуке. Новый «облачный» вариант поддерживается на большинстве основных настольных систем и, очевидно, в скором времени доберется до устройств iOS и Android.

От прошлой версии выжило несколько недостатков, так и не устраненных разработчиками Jolicloud. Например, в дате вашего рождения правильно выводятся год и месяц, но почему-то не день.

В главном окне Jolicloud всего два ряда значков. Для малых экранов это неплохо, но лучше было бы предусмотреть варианты выбора. То же относится к разрешению экрана и размерам значков – это позволило бы продвинуть дистрибутив на более широкий класс ноутбуков и машин, подсоединенных к Интернет.

Борцы за свободу ПО будут удручены тем, что, хотя основу дистрибутива составляет свободное ПО, интерфейс и «облачные» компоненты распространяются как под свободными, так и под проприетарными лицензиями. Обычному пользователю от этого ни холодно ни жарко, но Ричард Столлмен такое не похвалит.

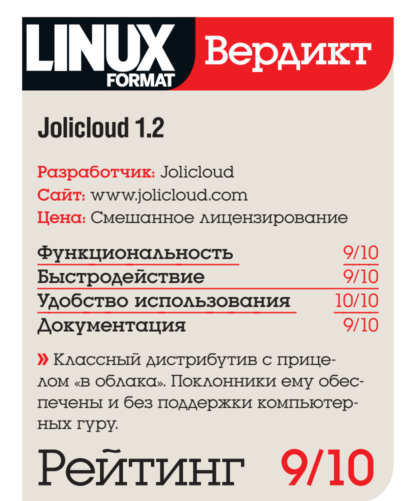
Несмотря на мелкие недостатки, дистрибутив за последние девять месяцев сумел дорасти до присуждения нашей награды Top Stuff: учтены новые социальные функции, улучшена поддержка аппаратного обеспечения, и видна забота об удобстве использования. Любой, кому попадает в руки устройство с этим дистрибутивом, осваивается почти мгновенно: здесь все «просто работает». LXF



Свойства навскидку

Социальные связи
Общайтесь с единомышленниками; узнавайте о новых приложениях.

Облака синхронны
Синхронизировать компьютеры и учетные записи онлайн — пара пустяков.



LINUX FORMAT Вердикт

Jolicloud 1.2

Разработчик: Jolicloud
Сайт: www.jolicloud.com
Цена: Смешанное лицензирование

Функциональность	9/10
Быстродействие	9/10
Удобство использования	10/10
Документация	9/10

» Классный дистрибутив с прицелом «в облака». Поклонники ему обеспечены и без поддержки компьютерных гуру.

Рейтинг 9/10

Spotify



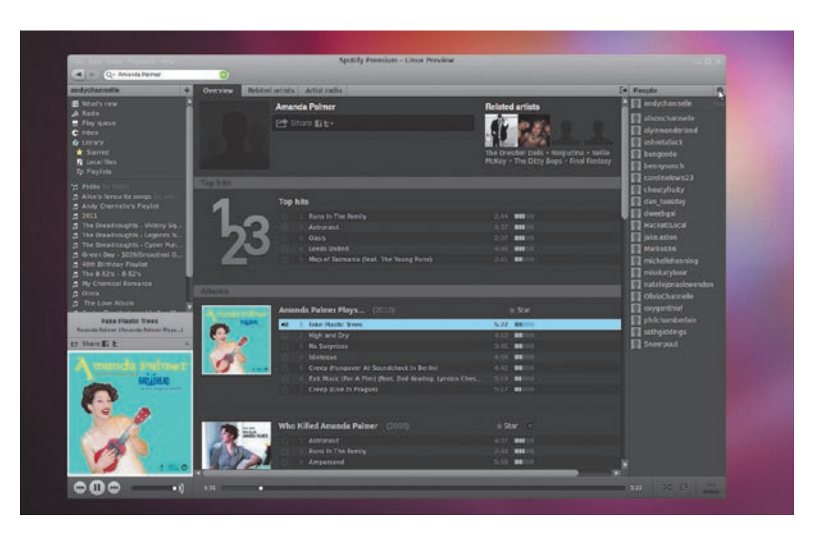
Наконец-то появился родной Linux-клиент для музыкальной службы Spotify. **Энди Ченнел** решил проверить, всегда ли родной значит лучший.

Вкратце

» Персональная служба музыкальной трансляции с бесплатной версией и тарифными планами месячной подписки. См. также: Last.fm и Rhapsody.

Когда мне повезло стать бета-тестером Spotify, я был изумлен. Поразили меня две вещи: а) по нажатию Play музыка сразу же начинала играть, без всякой паузы на загрузку исходной порции, и б) как на это пошла консервативная музыкальная индустрия? За несколько лет каталог разросся, миллионы людей ежемесячно вносят наличные за версию без рекламы, а пользователи Linux получили собственный клиент. До него приходилось либо обращаться к *Wine*, мирясь с его недостатками (например, монопольным захватом вашей звуковой системы), либо идти на поклон к Windows.

Пока что нет действенного способа избавиться от визуальной рекламы, которая сопровождает надоедливую звуковую, поэтому пользователям Linux придется ежемесячно платить £4,99 за уровень Unlimited или £9,99 за уровень Premium. Есть двоичные файлы для Fedora, Ubuntu и... все, пожалуй. Возможно, удастся установить RPM-пакет на SUSE или Mandriva, а Deb сработает на других потомках Debian, но успех не гарантируется.



› Пользователей Premium не мучают рекламой и выдают им музыку раньше других.

> «Ну как на это пошла консервативная музыкальная индустрия?»

Вообще-то и установка готовых двоичных пакетов не обошлась без препятствий. Например, версия Fedora не вписалась в Gnome, заставив искать приложение, а команда **apt** в Ubuntu норовила добавить несуществующий репозиторий исходного кода. Обе проблемы решились быстро, и через несколько секунд после нажатия кнопки Install мы уже искали в библиотеках старую добрую панк-музыку.

Служба и ПО

Следует раздельно учитывать два аспекта Spotify: службу и ПО. Для любителей музыки эта служба сущий клад. Заплатить 4,99 в месяц за неограниченный доступ к обширнейшей музыкальной библиотеке совсем не накладно, да и 9,99 фунта за полный смартфон музыки – тоже немного. ПО вполне сравнимо с версиями для Windows и Mac: обмен в социальных сетях, коллективное составление списков воспроизведения и доступ к локальным файлам – все на месте и работает не хуже, чем на других версиях.

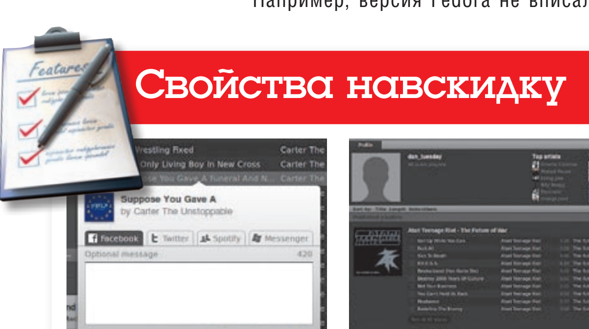
Недостатки тоже есть, но мелкие – например, не отображаются фотографии профиля для контактов Facebook – и не портят общего приятного впечатления от ПО. Неприятностей, характерных для версии *Wine* (захват аудиофокуса у других приложений, неряшливые следы на рабочем столе при перерисовке...), здесь нет, поэтому тот клиент можно отставить.

По быстродействию данная версия и вариант для Windows (на Windows или через *Wine*) почти не отличаются, а главное – музыка начинает играть сразу после нажатия кнопки Play. Некоторым пользователям слышали перебои в первые секунды воспроизведения. Нам не удалось повторить этот подвиг, даже наводнив систему и сеть потоками информации; возможно, дело было в звуковой карте.

Если вас устраивает бесплатная версия с рекламой, вариант для *Wine* по-прежнему доступен и действует неплохо. Однако, немного потратившись, можно получить гораздо больше удовольствия и убедиться в том, что коммерческие услуги для Linux стоят вложенных денег.

Лучшее ПО для работы с музыкальными библиотеками? Вероятно. Стоит запрошенных денег? Определенно. Конкуренты и в сравнение не идут. LXF

Свойства навскидку



Поделитесь мелодией
Щелкните правой кнопкой на композиции, альбоме или исполнителе и нажмите кнопку Share для отправки в свою социальную сеть.



Будьте на связи
Браузер People (здесь – без картинок профиля) показывает, что в данный момент слушают ваши друзья и члены семьи.



Spotify

Разработчик: Spotify Ltd
Сайт: www.spotify.com
Цена: Бесплатно/£4,99/£9,99

Функциональность	9/10
Быстродействие	10/10
Удобство использования	9/10
Оправданность цены	10/10

» У любителей музыки Spotify вызывает слезы счастья. Да, вот так все здорово.


Рейтинг 9/10

UTUTO XS 2011.03

В мире не так много «полностью» свободных дистрибутивов Linux. **Игорь Штомпель** рассматривает новый релиз одного из них.

Вкратце

» **UTUTO полностью соответствует «Рекомендациям для свободных дистрибутивных систем GNU/Linux». См. также BLAG, Dynebolic, gNewSense, Trisquel, Venenux.**

UTUTO – «полностью» свободная операционная система, признанная Фондом свободного ПО (FSF) (см. список свободных ОС на сайте GNU: http://www.gnu.org/distros/free-distros.ru.html); она основана на Gentoo. UTUTO – разработка Национального университета, находящегося в г. Сальта (Universidad Nacional de Salta, Аргентина). Название дистрибутив получил в честь шустрой местной ящерицы – она красуется и на официальном сайте, и на заставках ОС.

Проприетарных компонентов в UTUTO 2011 нет – только свободное ПО. Так, вместо привычного ядра Linux, как в большинстве дистрибутивов, UTUTO 2011.03 трудится на базе обновленного «полностью» свободного ядра – Linux-libre 2.6.37.1.

Основными новшествами предыдущего релиза, UTUTO XS 2011, были переход на KDE 4.6.1, *GCC 4.5.2*, *Glibc 2.13*, *Qt 4.7.2*. Кроме того, появилась возможность установки ОС из режима LiveCD и улучшена поддержка оборудования, а также сети (Ethernet, 3G, Bluetooth и Wireless). Апрельский релиз 2011.03 улучшил поддержку Ipv4/Ipv6 и использование оперативной памяти, а также оптимизацию установки на устаревшие машины. А на смену *OpenOffice.org* пришел *LibreOffice*.

Увы, документация по UTUTO оставляет желать лучшего, а среди доступных локализаций – только испанский язык.

Установка

Процесс установки своеобразен. Так, для запуска процесса установки введите в окне поиска главного меню instalar и запустите Instalar UTUTO XS. После этого запустится *GParted*, где будет необходимо создать как минимум два раздела – основной, на который планируется установить UTUTO, и раздел подкачки – swap. Далее процесс установки проводится с помощью web-интерфейса, запущенного в *Midori*. Вам останутся стандартные шаги: выбрать раздел, местоположение и т. д.



› **Рабочий стол UTUTO XS 2011.03.**

Рабочий стол

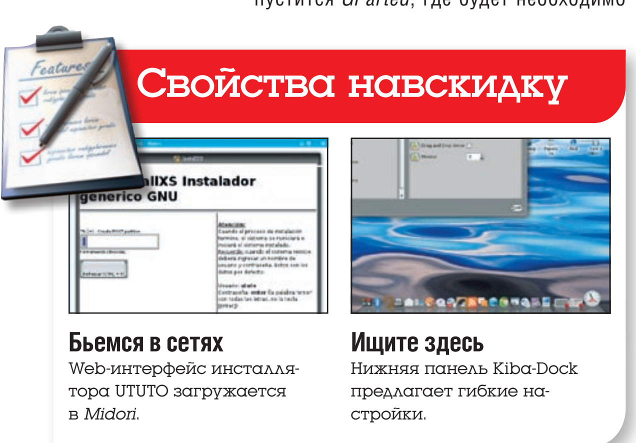
Разработчики предлагают пользователю на выбор GNOME, KDE, *Xfce* и *LXDE*. Это достаточно щедро. А запустить ОС UTUTU 2011 можно или с LiveDVD, или с USB-носителя; возможна установка на жесткий диск. Имя пользователя будетututo, пароль – enter (слово, а не нажатие кнопки).

Рабочий стол по умолчанию – KDE 4.6.1. Организация его необычна. Разработчики поместили главную панель вверху экрана – она занимает примерно его половину. Кроме того, на рабочем столе размещены два виджета – Reloj analogico и UTUTO XS GNU System. Первый – аналоговые часы, а второй открывает для доступа домашнюю папку, Параметры системы, Mozilla Thunderbird (клиент электронной почты по умолчанию), *Kopete* (клиент обмена мгновенными сообщениями по умолчанию), web-браузер *Chromium*, быстрый web-браузер *Midori*, корзину и каталог Сеть, или выполняет переход в спящий режим/выключение компьютера.

Нижнюю часть экрана занимает панель Kiba-Dock, с ярлыками множества приложений для быстрого запуска. Здесь и экранная клавиатура, и настройка Bluetoth, и зачем-то опять Параметры системы, окно, которое можно открыть с помощью виджета UTUTO XS GNU System, настройки печати и много другое.

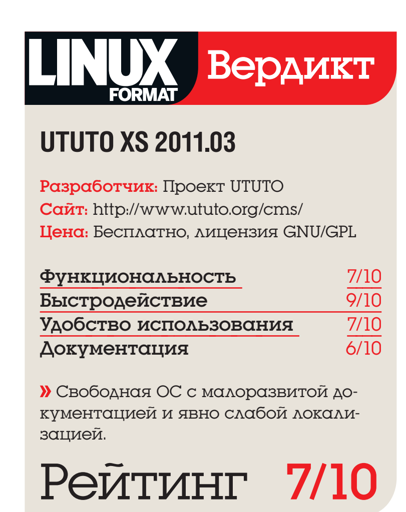
В целом, UTUTO производит хорошее впечатление. Но, на наш взгляд, проекту не мешало бы улучшить документацию, а также расширить локализации – ведь его ОС позиционируется как «полностью» свободная, а подобных не так уж и много. LXF

Свойства навскидку



Бьемся в сетях
Web-интерфейс инсталлятора UTUTO загружается в *Midori*.

Ищите здесь
Нижняя панель Kiba-Dock предлагает гибкие настройки.



UTUTO XS 2011.03

Разработчик: Проект UTUTO
Сайт: http://www.ututo.org/cms/
Цена: Бесплатно, лицензия GNU/GPL

Функциональность	7/10
Быстродействие	9/10
Удобство использования	7/10
Документация	6/10

» Свободная ОС с малоразвитой документацией и явно слабой локализацией.

Рейтинг 7/10

Qnap 219P+ против Synology DS211+

Грэм Моррисон столкнул лбами два современных мощных сетевых накопителя в бескомпромиссной схватке.

Вкратце

» Два сетевых накопителя на Linux-системе, предназначенные для дома или офиса.

Сетевые накопители на Linux-системах – крупный бизнес благодаря умеренным ценам этих богатых функциями устройств. Новейшие 219P+ от Qnap и DS211+ от Synology нацелены на один и тот же сегмент рынка и сравнимы по цене и техническим характеристикам. В обоих есть место под установку двух приводов (каждый емкостью до 3 ТБ), процессоры тактовой частотой 1,6 ГГц, ОЗУ на 512 МБ и специализированная версия Linux.

Установка

Оба NAS поставляются без приводов, и сперва придется их вставлять. В отличие от более дешевых накопителей, выпускаемых этими компаниями, здесь имеются кассеты для приводов, менять которые можно не отключая питание. Доступ к кассетам открывается с передней панели устройств. Пластмассовые кассеты Synology находятся под декоративной накладкой, которая крепится резиновыми держателями. Кассеты легко извлекаются, к ним крепятся приводы – и все это устанавливается обратно в корпус. Пластиковые язычки для 3,5-дюймовых приводов играют роль подвески, обеспечивая гашение вибрации и шума. Кассеты Qnap – металлические, и выглядят более солидно. Потянув верхнюю защелку, можно извлечь из корпуса весь отсек. Закрепляем 3,5- или 2,5-дюймовый привод винтами и вставляем обратно в корпус. Механическая часть Qnap проработана более основательно и дает ему конструктивное преимущество.

Роли меняются, когда дело доходит до установки встроенного ПО в новые накопители. На оба устройства ОС устанавливается либо в режиме загрузки, либо с прилагаемого CD. Но только у Synology есть Linux-инструмент для сетевого поиска машины и управления процессом установки. Qnap ставит вас перед выбором: либо воевать с инструкциями в командной строке, либо звонить знакомому владельцу Windows или Mac OS X. Непростительно для бренда, столько лет работающего с Linux.

Существенно различаются и пользовательские web-интерфейсы. Все устройства Synology значительно выиграли в результате обновления от версии 3.0: панель настройки страничного режима уступила место превосходной web-системе на основе HTML5. Теперь в одном из окон можно запустить музыкальный плейер, одновременно занимаясь настройкой сетевых параметров и читая превосходную документацию – и все это в пределах нового web-интерфейса.

Переход на Qnap выглядит возвратом в средневековье. Здесь web-интерфейс построен на меню и панелях, и хотя ничего плохого в этом нет, выигрыш в интуитивности и простоте использования бесспорно за Synology. Впрочем, незатейливый интерфейс Qnap открывает доступ к большему количеству функций.

Функции

Как и полагается приличным Linux-системам, оба накопителя оснащены такими функциями, которые большинству из нас никогда не понадобятся. С помощью стандартного интерфейса можно настроить их на использование в качестве роутеров, серверов печати и FTP и даже мониторов безопасности с использованием web-камер. Оба устройства обеспечивают общий доступ по протоколам



› У накопителя Qnap более прочный корпус, а также металлические кассеты для крепления двух приводов.

› Устройство от Synology выглядит менее профессионально, зато снабжено кард-ридером.

Пакеты и индивидуальная настройка

Замечательно, что оба устройства работают на достижимой версии Linux. Доступ возможен по протоколам Telnet и SSH, нужно только активировать его через интерфейс управления. Можно устанавливать приложения из пакетов, специально сформированных для каждого устройства. Synology тут выигрывает по удобству: хотя пакеты Qnap в виду проще, установка их более сложна и затейлива. На обе машины можно установить *Squeezebox* от Logitech, загрузив пакет и установив его с помощью графической оболочки. Нам удалось даже установить модуль *Spotify* на Qnap, поскольку там имеется библиотека поддержки ARM. К сожалению, использование процессора PowerPC не позволило проделать этот трюк на Synology, но это можно исправить.

Для Qnap доступно больше пакетов, чем для Synology, и *Optware*, систему управления пакетами IPKG, гораздо проще установить (по крайней мере, в теории). Пакеты подготовлены к этому процессу заранее, но нам все же пришлось поработать с командной строкой для установки переменных среды и редактирования скриптов запуска и завершения. Официальная поддержка этих функций для Synology не афишируется, но управлять загрузочным скриптом с помощью приложения стороннего производителя здесь гораздо проще. В любом случае после установки IPKG можно загружать сотни специально сформированных пакетов, от текстового редактора *Nano* до сервера *Subversion*.



› Приложений для Qnap немало, вот только установить их не так-то просто.

Samba и NFS, установку удаленных дисков iSCSI и права для пользователей и групп – для ограничения доступа к сети. В силу «домашней» направленности, устройства оснащены серверами для трансляции музыки, видеопрограмм и фотографий

> «Процессоры обоих – на 1,6 ГГц, много быстрее среднего.»

на устройства типа UPnP. Synology делает это через собственный интерфейс, а Qnap применяет стороннее приложение *TwonkyMedia* (полный обзор см. **LXF96**). Нам показался удобнее подход Synology, хотя и *TwonkyMedia* работает неплохо.

Оба устройства включают приложения для поддержки платформ Android и iPhone, предоставляющие доступ к музыкальным файлам, фотографиям и даже web-камерам прямо с ладони. Мы пробовали эти средства на iPhone – результаты плачевные. В пределах локальной сети система действует неплохо, но когда доходит до NAT и неустойчивых 3G-подключений, передача фотографий на накопитель или трансляция музыки на телефон становится проблематичной. По сути эти приложения очень интересны и обладают значительным потенциалом, но пока Qnap и Synology не добьются их стабильности, пользы от них немного. Верх здесь одерживает Synology – только потому, что в нем работа с музыкой и фотографиями выполняется отдельными приложениями.

Быстродействие

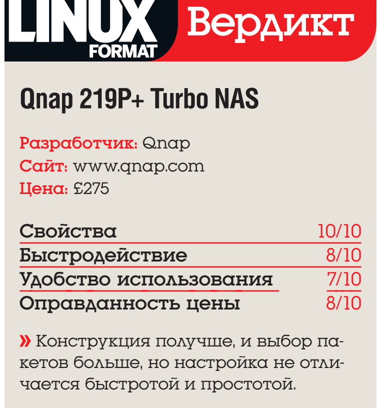
Процессоры обоих накопителей рассчитаны на тактовую частоту 1,6 ГГц (хотя и с различными архитектурами) – это много быстрее, чем у среднего NAS. В результате у обоих прекрасная производительность – особенно при копировании файлов, обработке крупных перечней медиа-файлов и работе с графическими оболочками проигрывателей. Вы это сразу заметите, запустив на любом из этих устройств *SqueezeCenter* и сравнив их с накопителем на процессоре с частотой 1 ГГц – на новом устройстве страница загрузится вдвое быстрее. В плане сетевой производительности обе машины лучше работали с NFS, чем с SMB, и Synology в большинстве случаев опережал Qnap. Например, скорость считывания одиночного крупного файла по протоколу SMB у Synology составляла 35 Мбит/с, а у Qnap 29 – Мбит/с. Однако в разных дистрибутивах получались разные результаты – то есть выбор дистрибутива может иметь большее значение, чем выбор модели накопителя, из-за различий в реализации *Samba*.

Затем мы испытали оба устройства, установив на них *WordPress*, что подразумевает работу со связкой *Apache*, PHP и *MySQL*. На Qnap этот процесс был чуть проще благодаря графической программе *PhpMyAdmin*. Кроме того, навигация в файловой системе показалась нам более удобной – хотя это все та же знакомая процедура, как на Linux-машине. В любом случае, оба устройства – прекрасная альтернатива серверам. Мы не стали бы рекомендовать их для web-хостинга, но в кругу семьи, друзей и коллег они вполне приемлемы.

Хотя Synology в большинстве сравнительных испытаний выходил победителем, ваш выбор зависит от конкретных потребностей – Qnap тоже по-своему хорош. Если вам нужно простое устройство, легко управляемое с Linux-компьютера, выбирайте Synology. Если нужна машина с более глубокими возможностями настройки и лучшей поддержкой ПО сторонних приложений, берите Qnap. Так и так деньги будут потрачены не зря. LXF



› Интерфейс Synology выше всяких похвал.



LINUX FORMAT Вердикт

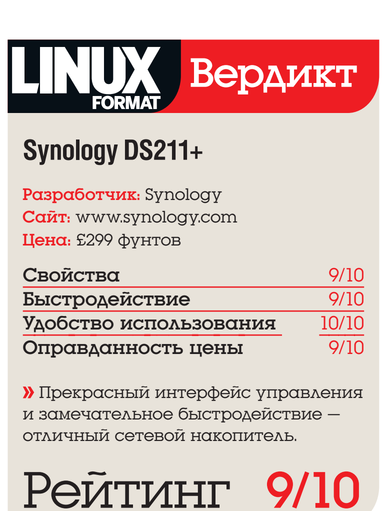
Qnap 219P+ Turbo NAS

Разработчик: Qnap
Сайт: www.qnap.com
Цена: £275

Свойства	10/10
Быстродействие	8/10
Удобство использования	7/10
Оправданность цены	8/10

» Конструкция получше, и выбор пакетов больше, но настройка не отличается быстротой и простотой.

Рейтинг 8/10



LINUX FORMAT Вердикт

Synology DS211+

Разработчик: Synology
Сайт: www.synology.com
Цена: £299 фунтов

Свойства	9/10
Быстродействие	9/10
Удобство использования	10/10
Оправданность цены	9/10

» Прекрасный интерфейс управления и замечательное быстродействие – отличный сетевой накопитель.

Рейтинг 9/10

Сравнение



Каждый месяц мы сравниваем тысячи программ – а вы можете отдыхать!

LINUX FORMAT на DVD
» Amarok 2.4.0
» Banshee 1.8.1
» Rhythmbox 0.13.3
» Songbird 1.10

Проигрыватели

Бим-бом! **Эндрю Грегори** ищет, чем бы порвать барабанные перепонки.



Про наш тест...

Так как *Amarok* глубоко интегрирован с KDE, мы тестировали его на этом рабочем столе, а *Rhythmbox* и *Banshee*, справедливости ради, тестировали в Gnome. *VLC* и *Songbird* тестировались на обоих рабочих столах. Мы ищем проигрыватель, который хорошо выглядит, воспроизводит музыку в максимальном количестве форматов, дает доступ к интернет-радиостанциям, таким как LastFM, а также позволяет покупать музыку в интернет-магазинах. Мы одолжили iPod, чтобы посмотреть, хорошо ли они с ним сработаются, а также протестировали всех на нашем 8-ГБ MP3-плейере Creative Zen.

Мы используем смесь MP3 и OGG – одни мы скачали с Vinyl Plus и Amazon, другие оцифровали сами. Такая разношерстная коллекция нужна для проверки способности приложений к распознаванию альбомов, и, возможно, даже дожмет нас до того, что мы когда-нибудь приведем их в порядок.

Музыка... собирает людей... вместе... как Мадонна незабываемо пела в своем хите 2000 года. И, судя по невероятно прибыльным продажам, это не безосновательно. Но если раньше для сохранения звуковых колебаний нужно было гравировать восковые цилиндры, дети наших дней обращаются к цифровым форматам, таким как Ogg, FLAC и MP3.

Большинство настольных дистрибутивов предлагают как минимум один плейер, который многие люди и не думают менять. Но поскольку Ubuntu 11.04 не включила старое приложение *Rhythmbox*, выбрав вместо него *Banshee*, наше любопытство было задето: какой из них лучший в Linux?

Мы составили список из трех, которые, по нашему мнению, все должны попробовать: *Rhythmbox*, плейер по умолчанию для Gnome; *Amarok*, плейер по умолчанию в KDE; и *Banshee*, новый плейер, который вытеснил *Rhythmbox* на Ubuntu. К ним мы добавили *Songbird*, музыкальный плейер, построенный на коде Mozilla, и доблестный *VLC*, который проигрывает всевозможные медиа-форматы на различных операционных системах с 1996 года.

Прямое сравнение таких разнородных программ может показаться несправедливым – скелетный плейер вроде *VLC* конкурирует за иную демографию, чем приложения с прибамбасами, подобные *Amarok* или *Banshee* – поэтому мы разбили Сравнение на категории.

> «Дети наших дней обращаются к форматам Ogg, FLAC и MP3.»

Оцифровка и форматы

Хорошо ли они справляются с проигрыванием вашей музыки?

Вы, наверное, воображаете, что, купив CD, имеете право слушать его любым удобным для вас способом. Как бы не так: все, что у вас есть, это лицензированная музыка, которая хранится на компакт-диске.

Музыкальные компании норовят ограничить варианты устройств, на каких его можно прослушать, и количество слушающих с вами. Они также хотят пресечь преобразование вами музыки в различные форматы, да и на форматы часто накладываются ограничения.

VLC, например, позволяет воспроизводить DVD, защищенные системой шифрования CSS, тем нарушая Digital Millennium Copyright Act США. Вот почему нам нельзя поместить *VLC* на **LXFDVD**. И если у вас проблема с кодеками, это почти наверняка проблема вашего дистрибутива. Пользователи, пытающиеся играть MP3 на *Amarok*, будут удачливее, если у них Linux Mint, а не gNewSense.

Это осложняет задачу разработчиков, и решение, к которому пришло большинство проектов – разделение поддержки форматов файлов на отдельные модули, устанавливаемые по усмотрению пользователя в соответствии с местным законодательством. *Banshee*, *Rhythmbox* и *Songbird* все работают на движке *Gstreamer*, применяющем систему модулей расширения для поддержки огромного разнообразия форматов. Если у вас проблемы с проигрыванием файлов, вы, вероятно, не установили правильный модуль *Gstreamer*.

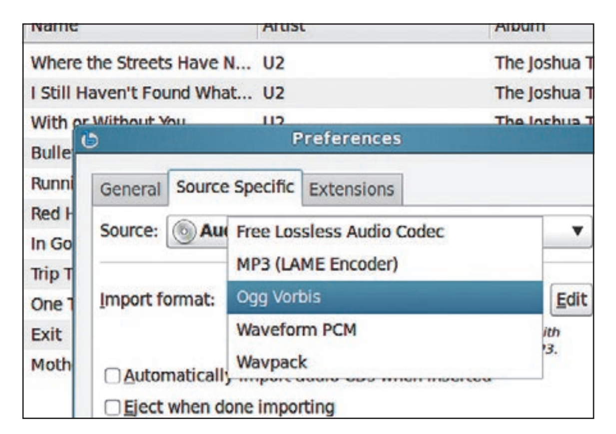
Общая радость

Amarok по умолчанию использует движок *Phonon*, не позволяющий воспроизводить файлы с ограничениями DRM. Он также почти не дал нам проиграть MP3-файлы. Если вы пробовали использовать *Amarok* на рабочем столе Gnome, вы, наверное, увидели сообщение об ошибке о том, что он не проигрывает MP3. Это проблема комплектации, а не самого *Amarok*, хотя возможность исправления следовало бы объяснить понятнее (нам помог Google). MP3 прекрасно проигрываются на *KDE*, как и Ogg, FLAC, AAC, WAV, WMA и WavPack.

Кросс-платформенный *VLC* в деле уже давно, и можно ожидать, что он в состоянии обрабатывать все.

Ничего не может быть проще, чем оцифровка CD в *Banshee*. В отличие от отдельных утилит для оцифровки, типа *Sound Juicer*, треки автоматически добавляются в библиотеку. В *Rhythmbox* это не сложнее, но мы нашли данную функцию ненадежной – рекомендуем прибегнуть к *Sound Juicer*.

Amarok использует специальный инструмент для оцифровки компакт-дисков. Нужная вам функция – Извлечь Digital Audio с K3b. *Songbird*, похоже, не знает способа получения аудио из вашей коллекции компакт-дисков (хотя имеется модуль для добавления этой функции). *VLC* такое умеет, но опции спрятаны под Media > Преобразовать/Сохранить.



› По умолчанию *Banshee* оцифровывает CD в Ogg Vorbis, но предлагает четыре других формата на выбор.



Вердикт

Amarok ★★★★☆

Rhythmbox ★★★★☆

VLC ★★★★☆

Songbird ★★☆☆☆

Banshee ★★★★★

» Amarok требует отдельного приложения, но он работает, и мы не будем брюзжать.

Поддержка устройств

Пользователи Linux тоже заслуживают красивых гаджетов.

Что бы вы ни думали о закрытой бизнес-модели Apple и ее навязчивом контроле, они все равно блестяще производят прекрасные девайсы.

Мы не ожидали большого успеха с iPod Nano, но *Banshee*, *Amarok* и *Rhythmbox* доказали обратное, распознавая устройство идеально, импортируя музыку в приложения и загружая обложки. Синхронизация с iPod прошла также легко: это делается щелчком правой кнопки мыши на иконке iPod в *Rhythmbox* и *Amarok*, или через меню настройки, появляющееся, когда вы выделяете устройство в панели источников в *Banshee*.

VLC просто открывает окно, отображающее все содержимое iPod (в том числе информацию календаря, файлы настройки и звуковой сигнал). Он пропускает файлы, которые ему не распознать, и начинает играть с первого распознанного – который оказался пошлой мелодией диско. Не хотелось бы, чтобы такое происходило при каждом обращении к iPod.

Мы также протестировали 8-ГБ Creative Zen. Оно довольно примитивное – мы использовали его как запоминающее устройство USB и управляли файлами через файловый менеджер *Nautilus*, а не через специальное приложение для воспроизведения музыки – но *Rhythmbox* и *Banshee* работали с ним не хуже, чем с iPod. В *Amarok* устройства также появились в панели источников, а его содержание отображается так же, как это было в *Banshee* и *Rhythmbox*, хотя мы столкнулись-таки с проблемами в версии, установленной на рабочем столе Gnome.

А вот *VLC* совсем не смог достучаться до устройства Creative, хотя и распознал iPod. *Songbird* не работал ни с одним устройством – это приложение скорее для Интернет, чем для реальной жизни.



› *Rhythmbox*, как и *Amarok*, *VLC* и *Banshee*, распознал наше тестовое устройство iPod.

> «Banshee, Amarok и Rythmbox распознали iPod отлично.»



Вердикт

Amarok ★★★★★

Rhythmbox ★★★★★

VLC ★★★☆☆

Songbird ★☆☆☆☆

Banshee ★★★★★

» Хорошие показатели у трех лидеров. Стив Джобс будет доволен.

Интерфейс

Важно ли, чтобы музыка выглядела симпатично? Конечно, важно!

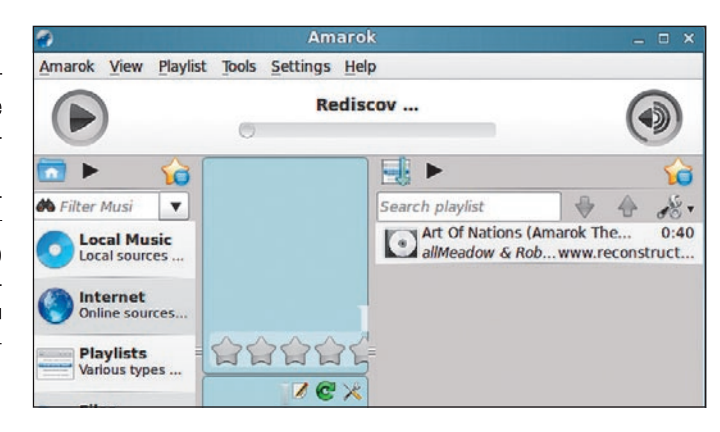
Ваши воззрения на вид медиа-проигрывателя, вероятно, зависит от способа его использования. Если вы хотите сляпать плей-лист, а затем свернуть его в лоток и забыть о нем, вы вряд ли по достоинству оцените графические тонкости, в отличие от тех, кто хочет извлечь максимум ощущений. По крайней мере, мы так думали. На практике, хорошо продуманный интерфейс имеет смысл независимо от того, сколько времени вы сидите перед компьютером – можно даже утверждать, что когорта «поиграл и забыл» требует более удобного интерфейса, потому что они тратят меньше времени на изучение использования ПО и т. д. и должны быть в состоянии быстро отыскать функции, которые им действительно нужны. Приложения в этом тесте различаются по своим подходам, и вы обязательно найдете то, что хотите.

Amarok



При первом запуске *Amarok* вы увидите экран с правой стороны. Там будет много пустых и серых полей – несмотря на это, слоган приложения «Откройте для себя музыку» укорочен до «Откро...». Ничто, кажется, никуда не вмещается; а поскольку иконок явный перебор, не ясно, с чего начинать.

Когда вы заполните свою музыкальную библиотеку и расширите окно приложения до разумных размеров, смысла станет больше. *Amarok* использует довольно традиционный трехпанельный вид, показывающий (слева направо) источник музыки; информацию, которая меняется в зависимости от воспроизведения песни; список воспроизведения. Цена высокой настраиваемости в *Amarok* – неопрятный внешний вид, а уведомления, появляющиеся при отслеживании изменений, на диво уродливы.



Banshee



По умолчанию у *Banshee* самая большая панель для отображения альбомов с обложками. Это увеличивает объем списка, но зато как приятно скопировать запись и полюбоваться обложками альбомов в автобусе по дороге домой, мысленно представляя, как наше новое приобретение звучит.

Интерфейс *GTK* прост и понятен. Если вы хотите играть песню, сообразить, что делать, будет легко. Меню достаточно доходчивы, так что можно выполнять и более сложные маневры: создать плей-лист, подключиться к интернет-радио или упорядочить большую коллекцию. Интерфейс *Banshee* до известной степени настраиваем, хотя опций куда меньше, чем в *Amarok*. Что и характерно для приложений Gnome, инструменты настройки скрыты от глаз, пока вы их не запросите – полная противоположность *Amarok*.



Музыкальные магазины в Сети

Платите за музыку онлайн – так велел Фергал Шарки [Faergal Sharkey].

Когда-то был всего один приличный Интернет-магазин музыки: разорительный iTunes, осложненный правами DRM. Как и следовало ожидать, с тех пор все улучшилось, и конкуренция ввела различные способы оплаты музыки.

Amarok и *Rhythmbox* оба предлагают сервисы Jamendo и Magnatune, дающие доступ к огромным коллекциям произведений малоизвестных авторов. Вы не найдете здесь плей-листа Radio One, но насладиться музыкой очень даже можно; контент передается бесплатно – просто сохраните музыку на жестком диске; или сделайте добровольное пожертвование. Правда, при нажатии на иконку Jamendo *Rhythmbox* нагреб столько музыки в нашу коллекцию, что сам же и повис. *VLC* работает с Jamendo намного лучше, разбивая результат на поддающиеся управлению категории.

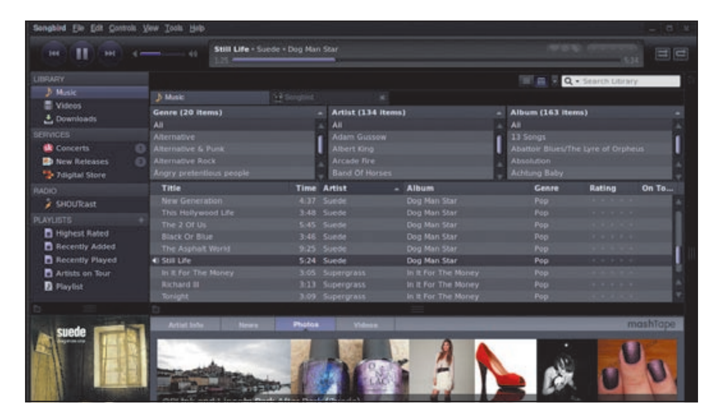
Лучшие услуги предоставляют *Banshee* (с Amazon) и *Songbird* (с 7digital). Оба эти коммерческих музыкальных магазина интегрированы с приложениями, и музыка и обложки альбомов в вашей коллекции появятся автоматически. 7digital имеет честь снабжать движком службу загрузки музыки Ubuntu One, и затмевает Amazon с точки зрения удобства прослушивания фрагментов ваших потенциальных покупок в интерфейсе *Songbird*, а не в модификации браузера, как Amazon делает с *Banshee*. Но Amazon предлагает более широкий выбор, так что почет и ему.



Вердикт

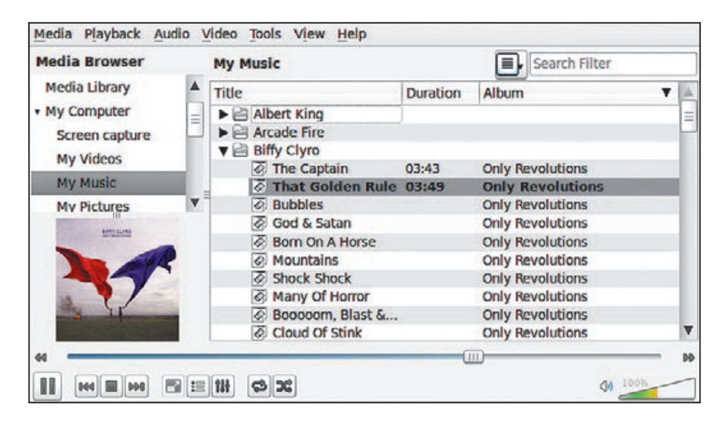
Amarok ★★★☆☆
Rhythmbox ★☆☆☆☆
VLC ★★★☆☆
Songbird ★★★★★
Banshee ★★★★★

» Banshee и Songbird выигрывают благодаря прагматизму.



Songbird ★★★★☆

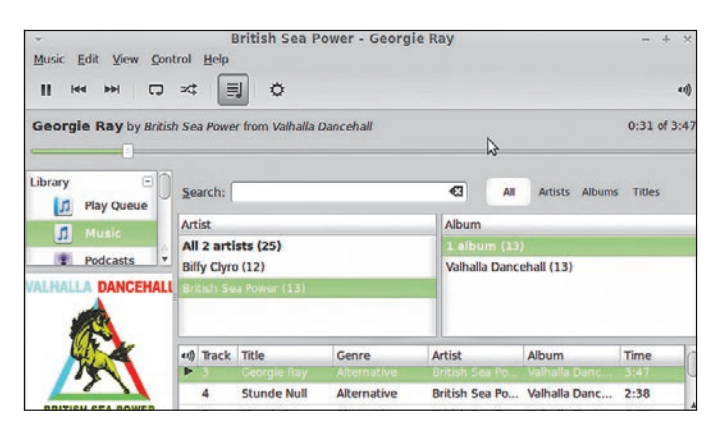
Songbird [*англ.* певчая птица] – забавная штука. Это web-браузер на базе кода Mozilla, и его внешний вид и поведение совершенно не похожи на традиционный музыкальный плейер. Главная панель показывает вашу музыкальную коллекцию, на панели слева вы выбираете источник медиа, а кнопки управления находятся в верхней части. Пока все обычно; но основная панель – на самом деле переодетое окно браузера! Нажмите на меню Справка на панели инструментов, и в новой вкладке откроется www.getsongbird.com. То же делают и другие приложения – все они захватывают содержимое из Интернет, чтобы добавить к проигрыванию вашей музыки; но *Songbird* отображает его на исходной web-странице. Набор скинов (здесь именуемых «перьями») позволяет подобрать внешний вид, хотя прочая настройка ограничена в опциях.



VLC ★☆☆☆☆

Власть имущие в команде *VLC* выбрали логотипом проекта полосатый знак дорожных работ, воспринимаемый как нечто в стадии доделки. Но может быть, мы зашли не с того конца и он знаменует утилитаризм, потому что в текущем интерфейсе *VLC* ни капли жира. Мы говорим «текущий», но *VLC* находится в разработке с 1996 (в терминах свободного ПО – примерно миллион лет), и за все это время разработчики упорно игнорировали все графические тонкости. Инструменты приходят и уходят; шейдеры, эффекты OpenGL и вращающиеся кубы обошли *VLC* стороной.

VLC дает немногим больше стандартного файлового браузера для навигации по музыке в обычном окне. Работать он работает, но это чистый функционализм.



Rhythmbox ★★★☆☆

Rhythmbox – проигрыватель музыки умолчанию в Gnome; он и выглядит как типичное приложение Gnome. Это и хорошо, и плохо. Хорошо – потому что он понятный и простой и в нем легко ориентироваться. Плохо – потому что малость скучновато. Крайняя левая панель отображает доступные источники музыки, а исполнитель, альбом и отдельные треки представлены в самой крупной панели.

Богатые метаданные, как в *Banshee*, *Amarok* и *Songbird*, здесь отсутствуют, и в результате *Rhythmbox* избегает хаоса. По части эстетики он напоминает улучшенную версию *VLC* – что неплохо, если вам просто нужна фоновая музычка – или, может быть, разработчики полагают, что незачем смотреть на экран с текстами песен и биографиями групп, когда есть дела поинтереснее?

Интернет-радио и подкасты

Радио – это звук спасения, радио очищает нацию.

Мы знаем, что повсюду есть подкасты и радиопотоки, но желаем, чтобы приложение помогало нам их найти. Ноль очков – у *Amarok*, который по нажатию на кнопку Подкасты в панели источников просто предлагает добавить RSS или Atom-каналы наших любимых подкастов. *Rhythmbox* в этом отношении ушел недалеко, но в Linux Mint к нему добавлена лента MintCast, помогающая стартовать. *Rhythmbox* также надбавляет очки за интеграцию с LastFM.

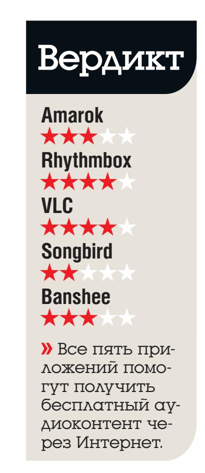
Когда вы подпишетесь на эту услугу и укажете «своих» исполнителей, LastFM будет транслировать поток их аудио прямо в *Rhythmbox*. *Rhythmbox* также выводит список потоковых радиостанций в гораздо более внятной форме, чем *Banshee* или *Amarok*.

Реализация LastFM в *Banshee* скудна, но он отлично подходит для подкастов. Он интегрирован с большим сайтом Miro, бесплатно делающим звуко- и видеозаписи, которые можно добавить в библиотеку.

VLC справляется неплохо, предлагая радиопотоки на выбор через *Icecast*. Хотя его поддержка подкастов не лучше, чем у *Amarok*.

Для приложения, построенного на web-браузере, *Songbird* разочаровывает. Значок Shoutcast есть, а способа просмотреть станции нет.

Songbird позволяет оценить текущий трек, «полюбить» его или заблокировать в потоке LastFM, но увы – проиграть LastFM мы очевидного способа не нашли.



Управление музыкой

Эти импорты с редкостного винила сами не организуются.

Для управления такой небольшой коллекцией, как наша, особых опций сортировки не требуется. Около 8 ГБ вроде и много, но если у вас музыки больше, эти альбомы скорее всего обречены затеряться в пространстве вашего жесткого диска. Если учесть, что одной из причин оцифровки музыкальной коллекции является удобство, вы довольно быстро поймете, что умение разыскать ваши любимые песни имеет первостепенную важность.

С этой точки зрения *VLC* замыкает список. Если там и есть способ организовать плей-лист, мы его упустили.

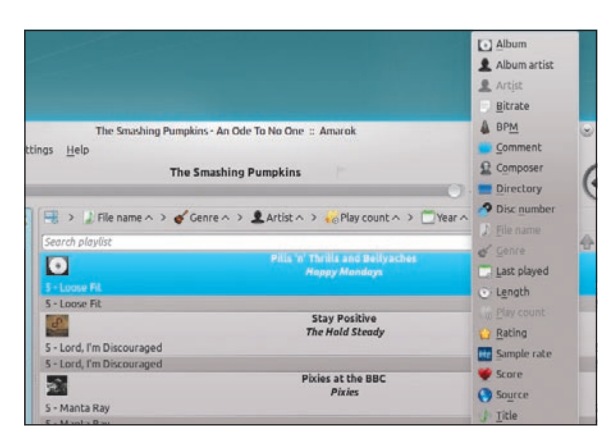
Rhythmbox умеет использовать номер дорожки, название, жанр, исполнитель, альбом и длительность песни как фильтры для сортировки треков в плей-листе (в настройках можно добавить другие опции), а также предлагает окно поиска. Этот подход интуитивен, но ограничен. Вы получите больше контроля в главном экране музыкальной библиотеки, где показаны панели источников, а также окна, отображающие жанры и исполнителей. Можно, например, выбрать «Элвис», затем «Джаз» в соответствующих окнах, и получить список всех джазовых альбомов – американский ответ Тому Джонсу.

Banshee похож на *Rhythmbox*, но использует окна для поиска по исполнителю и альбому, с соответствующим треком, показанным ниже. Разница минимальна, но мы предпочитаем подход *Rhythmbox*, потому что обложки в *Banshee* затрудняют поиск по большому количеству альбомов. *Songbird* и того лучше, с тремя панелями, которые помогают ограничить количество треков для поиска.

Умные плей-листы

Куда более впечатляет то, что *Banshee*, *Rhythmbox* и *Songbird* имеют «умные» плей-листы, которые автоматически порождают плей-листы согласно критериям, определенным пользователем. Например, имеется плей-лист «Недавно проигрывалось». В *Songbird* компилируются все аудиотреки, прослушанные за последние недели, но вы можете изменить критерии и подобрать для включения все, что вам хочется.

Здесь несомненно лидирует *Amarok*. Он предоставляет поиск в списке воспроизведения по исполнителям, альбомам, году и названию трека, и есть даже такие нестандартные критерии, как количество тактов в минуту, частота дискретизации и время последнего проигрывания. 22 опции фильтра помогут вам найти искомое, и допускается вложенный поиск, что облегчает извлечение (например) всех ваших треков Бетховена с высшим рейтингом на 127 тактов. Понимаем экзотичность подобного примера, но вдруг окажется, что вы DJ, который любит классическую музыку?



› Опции поиска *Amarok* огромны — смотрите, какие опции он вам дает!

Вы можете решить, много или мало информации отображать, чтобы помочь вам откопать желаемые музыкальные произведения, и есть поле поиска, где можно ввести запрос для поиска треков.



Модули расширения

Примочки для обогащения функциональности вашего приложения.

Firefox добился популярности благодаря изобилию модулей расширения. От апплетов погоды до показателей уровня террористической угрозы, вы обязательно найдете модуль, добавляющий нужную вам функциональность. И поскольку *Songbird* и *Firefox* – двоюродные братья, нечего и удивляться, что сайт *Songbird* предлагает кучу (сейчас – 593) дополнений, чтобы улучшить *Songbird* (в том числе дополнения для поддержки iPod – интересно, много ли «упущенной» функциональности переведено в модуль?).

У *Rhythmbox* – 57 модулей; наш любимец – Продолжить при перезагрузке, он верен своему названию и делает именно то, что вы и подумали. У *Banshee* 21 модуль, в том числе Cover Flow, который, несомненно, войдет в основной релиз, как только достигнет стабильности (на момент написания статьи он был все еще в альфа-версии).



› *Songbird* построен на коде *Mozilla*, и вы найдете для него тонны модулей расширения — есть даже и полезные.

Но нас интересует не столько количество модулей, сколько простота их получения для конечного пользователя. И тут главенствует *Amarok*.

> «Наш любимый модуль Amarok — Музыкальная викторина.»

Выбор пункта меню Инструменты > Менеджер сценариев > Получить больше Сценариев дает доступ к множеству радиостанций, песенных приложений, потоку изречений Корана и даже этому апофеозу бессмыслицы – индикатору настроения.

Наш любимый, пожалуй, Music Quiz – Музыкальная викторина; он проигрывает песни и задает вам всяческие вопросы про вашу коллекцию.

VLC тоже имеет интерактивный механизм дополнений, но без описаний, рейтингов и экранных снимков, как у *Amarok*.



Проигрыватели

Вердикт

Все рассмотренные здесь продукты имеют что предложить. *Songbird*, в частности, заслуживает упоминания как отличный проект. Он уже не поддерживается официально на платформе Linux, но это все равно отличное ПО, и стоит времени на знакомство.

Затем, есть и другой кандидат за все – *VLC*. Наткнувшись на файл, который не открывается, мы обратимся к *VLC*. Он работает на OS X и Windows, загружается в считанные секунды и потребляет минимум оперативной памяти; если вам когда-нибудь придется работать за заемным компьютером, можете не мириться с ужасным ресурсоедом *Windows Media Player*.

Ближе к вершине грызня обостряется. *Rhythmbox* необычайно удобен, но проигрывает в сравнении с *Banshee*, который чуть увереннее по всем аспектам. *Banshee* написан на *Mono*, открытой реализации языка C# от Microsoft, и если вас беспокоит сопутствующая патентная угроза, выбирайте *Rhythmbox*.

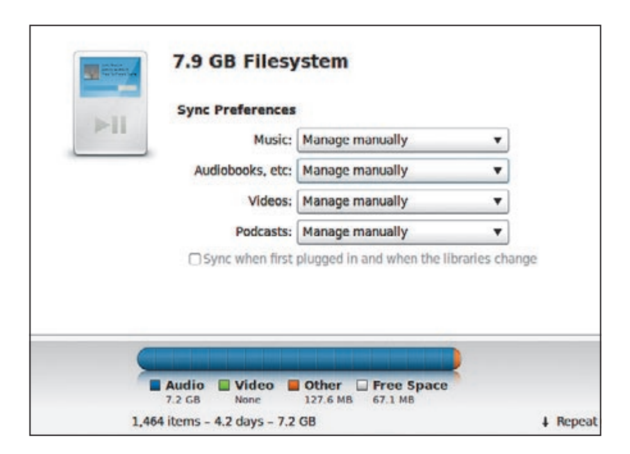
Кто кому волк

По одному приложению мы не смогли придти к заключению – это *Amarok*. Если у вас огромная коллекция музыки, которой нужно управлять причудливым образом, недоступным простым смертным, он, вероятно, будет для вас правильным выбором.

Amarok умеет то же, что и другие представленные здесь приложения, но его омрачает жуткий интерфейс, превращающий простейшие задачи в лабиринт догадок, проб и ошибок. Однако мы подозреваем, что единственным камнем преткновения является крутизна кривой обучения. Те, у кого терпение не лопнет, найдут в нем полный арсенал возможностей.

И остается только *Banshee*. Пусть это не понравится команде «Объявим бойкот Novell», но он нам всех милее. Благодаря несвободному хранилищу *Gstreamer*, он способен сыграть все форматы файлов, прекрасно работает с Amazon, хорошо выглядит, и это самое простое приложение в использовании.

Banshee – также единственное приложение, которое использует блестящее руководство-подкаст Миро, открывая мир бесплатного онлайн-обучения/болтовни про пони. Переход Ubuntu с *Rhythmbox* на *Banshee* приведет к нему много новых пользователей, и вполне заслуженно.



› Дни танцев с бубном вокруг *gtkpod* миновали — *Banshee* может все.



Сайт: http://banshee.fm **Лицензия:** MIT/X11 **Версия:** 1.8

» Banshee не делает ничего особенного сверх других, но как же с ним легко!



Сайт: www.rhythmbox.org **Лицензия:** GPL **Версия:** 0.13.3

» Rhythmbox – отличное приложение, но недостаточно отполированное.



Сайт: http://amarok.kde.org/ **Лицензия:** GPL **Версия:** 2.4.1 beta 1

» Масса возможностей по организации музыки, но изрядная головная боль при освоении.



Сайт: www.getsongbird.com **Лицензия:** GPL **Версия:** 1.7.2

» Отличный выбор для тех, кто страдает рассеянием внимания и любит внешний блеск.

Информация к размышлению

Пока мы не увидели, что Ник планирует включить *Clementine* в текущий выпуск HotPicks, мы весело резвились в альтернативной реальности, воображая, как мог бы сегодня выглядеть *Amarok*, не будь KDE 4.

Не хотелось бы перебегать дорогу Нику, но все же скажем, что этот потомок *Amarok* 1.4 показался нам миловидным. Есть также *Exaile*, *Amarok 1.4*, переписанный на *GTK*. Он ухитрился подвесить нашу систему Mint 10, и даже неясно, зачем он создан – медиа-приложений *GTK* полно и так.

Если вы слишком круты, чтобы снизойти до GUI, взгляните на *xmms2*, проигрыватель командной строки с простым для понимания синтаксисом. Или, если вы в восторге от *Firefox 4* и хотите, чтобы все в жизни добывалось через вкладки, можете попробовать *Guayadeque*, приложение под Gnome: внешне оно похоже на *Rhythmbox* и добавляет метаданные в стиле *Amarok*, уделяя больше внимания рейтингам и изображениям.

Возможно, самым перспективным выглядит проект *Songbird*. Ну да, мы и без того извели на него много журнальной площади, но если возобновится разработка клиента Linux, то он способен завоевать весь мир. LXF

Обратная связь

Отпугнет ли Mono вас от *Banshee*? Можно ли назвать *Rhythmbox* вымершим? Вы один из тех, кто обожает *Amarok*? Присылайте ваши мнения – по данному Сравнению и другие – на letters@linuxformat.ru.

GNOME 3.0 с нами!



Маянк Шарма представляет релиз, в корне меняющий рабочий стол.

Gnome 3 — первый главный релиз популярного окружения рабочего стола за последние восемь лет. Многие считают Gnome лицом Linux, так как большинство ранних дистрибутивов (и, конечно, много современных) принимали этот рабочий стол, чтобы выгодно украсить снаружи все более развивающееся ядро внутри. Gnome 3 радикально отличается от предыдущих релизов. У него лощеный интерфейс пользователя, с упором на практичность, и приглаженная кодовая база, которая порадует разработчиков приложений.

> «У него лощеный интерфейс с упором на практичность.»

Gnome Shell – новый интерфейс, который заменит несколько ключевых компонентов рабочего стола, таких как панель и меню, и повлияет на способ переключения задач, перемещения между окнами и многое другое. Наряду с *GTK 3* и оконным менеджером *Mutter*, основанным на *Clutter*, взамен *Metacity*, рабочий стол как единое целое прошел долгий путь с тех пор, как компоненты начали появляться в репозиториях многих дистрибутивов. Итак, без лишней суеты изложим все, что вам нужно знать о блистательном новом рабочем столе.

Разработка

Исследования, переосмысления и задержки, вошедшие в Gnome 3.

Рабочий стол Gnome 3 в деле совсем недавно. Впервые команда разработчиков объявила миру о зачинании Gnome 3 на Gnome Users And Developers European Conference (GUADEC) в Стамбуле в 2008 году. После конференции несколько программистов начали исследовать области в Gnome 2, требовавшие переделки ради более удобной работы.

Дискуссии привели к проведению встречи разработчиков Boston User Experience в октябре того же года. Винсент Унтц [Vincent Untz], бывший тогда в команде разработчиков, написал в блоге, что на конференции дизайнеры и разработчики «пытались забыть про текущий Gnome и понять, что может иметь смысл». Были рассмотрены несколько областей – управление окнами и файлами, виджеты рабочего стола – с дизайнерской точки зрения. В интервью *Gnome Journal*, Уильям Джон Мак-Канн [William Jon McCann], который разрабатывал несколько ключевых областей Gnome, отметил, что на встрече в Бостоне еще до ее начала приоритет был отдан дизайну.

Именно там был разработан первый макет Gnome 3. Он предварил несколько улучшений рабочего стола – изюминок Gnome 3, например, встроенный в рабочий стол поиск и графический интерфейс для перемещения окон между рабочими пространствами.

Расхождение во мнениях

Даже перед самым релизом Gnome 3 был узлом противоречий, самое известное из которых – отказ Ubuntu от него в пользу доморощенного интерфейса Unity. Разработчики Gnome не стали облегчать себе жизнь, проголосовав за настроенный рабочий стол, который не принуждает пользователя возиться с системными настройками. Большинство членов сообщества не были готовы к рабочему столу с аппаратным ускорением и урезанными опциями настроек. Но посидите немного в его интуитивном интерфейсе, осмотрите улучшенную систему оповещений, поиграйте с сенсорными значками рабочего стола, и, возможно, вы уже не уйдете на другой рабочий стол.

Gnome 3 – это только первый релиз самого главного переосмысления рабочего стола. Работа ведется, и если вы заметите заскоки, можно ожидать, что в следующих релизах их устранят.

Выход из оболочки

На той встрече разработчиков одной из ключевых позиций, которые решили переосмыслить, было управление окнами. В текущей версии Gnome такие задачи, как запуск приложений, управление окнами и переключение между задачами, обрабатываются в *Metacity*. Теперь это передается в *Gnome Shell*.

Говоря по-простому, *Gnome Shell* меняет визуальное представление о рабочем столе. Через шесть месяцев после старта проекта под началом Оуэна Тейлора [Owen Taylor], к руководству команды присоединился Уильям Джон Мак-Канн. «Первое, что я делал, начав работу над проектом *Gnome Shell*, это чтение. Я провел целый месяц, не занимаясь ничем, кроме сравнения пользовательских интерфейсов всех современных операционных систем для настольных компьютеров», говорит Мак-Канн.

Вместо обычной шлифовки существующего рабочего стола, разработчики выполнили дизайн после анализа основных моделей использования. Тейлор очертил вопросы, которые они хотели решить в новом рабочем столе: «Как обеспечить рабочий процесс, чтобы он помогал пользователю эффективно перемещаться между задачами, без грязной работы по управлению своими файлами и задачами? Как видоизменится рабочий стол, если пользователь большую часть времени проводит в web-браузере и программе для чата, а не в текстовом процессоре и электронных таблицах?»

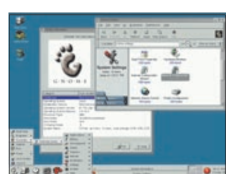
Алан Дэй [Allan Day], член команды разработчиков, добавляет: «Рабочий стол в Gnome 3 разработан так, что он будет узнаваем и прост в управлении на широком классе компьютерных устройств», и поэтому Gnome 3 предназначен для работы везде, от ограниченного пространства нетбука до ноутбуков с сенсорным экраном или настольных системах с несколькими ядрами.

Gnome Shell использует композицию окон, и в него встроен *Mutter,* менеджер окон на базе OpenGL, задействующий инструментарий *Clutter.* На саммите в Бостоне было несколько презентаций по использованию инструментов *Clutter* в ряде приложений Gnome. Именно там Фредерико Мена-Кинтеро [Frederico Mena-Quintero] представил свою утилиту управления файлами на основе журнала, которая сейчас называется *Zeitgeist* и является составной частью *Activity Journal* в Gnome 3.

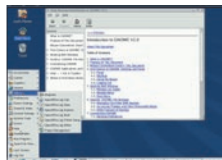
Было решено выпустить релиз Gnome 3 в сентябре 2010, но в июле того года команда отозвала релиз до марта 2011. К тому времени был выпущен Gnome 2.32, и версия 3 не ожидалась до апреля 2011 года.

»

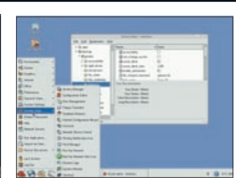
История Gnome



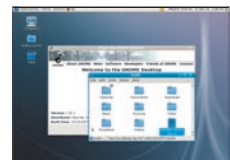
1.0 (март 1999)
Мигель де Икаса [Miguel de Icaza] объявил о первом главном релизе на Linux World Conference. Два года ушло на разработку; рабочий стол поддерживает перетаскивание и управление сеансами.



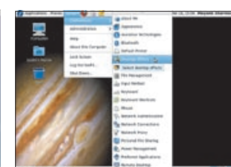
2.0 (июнь 2002)
Основанная на *GTK 2*, эта версия содержит множество улучшений: сглаженный текст, лучшая поддержка интернационализации, новые функции для удобства и видимые сдвиги в интерфейсе.



2.6 (март 2006)
Несколько улучшений в файловом менеджере *Nautilus,* включая новый режим Spatial, а также обновления *Control Center* и других базовых приложений.



2.20 (сентябрь 2007)
Gnome отмечает десятилетнюю годовщину добавлением новых утилит шифрования файлов, которые интегрированы в Gnome keyring; дебютировал автоматический установщик кодеков для мультимедиа.



2.28 (сентябрь 2009)
В релиз включен новый модуль, облегчающий управление устройствами Bluetooth, *Epiphany* переходит на *WebKit*, а *Empathy* переработан. Встроен просмотрщик удаленного рабочего стола *Vino.*



2.32 (сентябрь 2010)
Последний релиз из ветки Gnome 2.0 содержит несколько небольших дополнений и улучшений, но не являет особо волнующих функций, поскольку большая часть разработки перешла в Gnome 3.

Тур с остановками по рабочему столу Gnome 3

Быстрое знакомство с изменениями, бросающимися в глаза.



1 **Панель:** Основной рабочий стол заметно преобразился и делает громадные успехи по части дружелюбия к новичкам (картинку получше см на обороте). Вам не обязательно быть ветераном Gnome, чтобы заметить, что вместо двух панелей на рабочем столе осталась одна – вверху.

В панели изменено расположение часов и значков уведомления, но меню для приложений, перехода и системы влились в единое меню Activities. Рядом с этим меню отображаются активные программы, там предлагаются задачи, специфические для каждого приложения, а также опции закрытия программы и открытия нового окна.

2 **Activities:** Здесь можно увидеть самую горячо обсуждаемую инновацию Gnome Shell. Activities предлагают различные способы переключиться с одного рода деятельности или задачи на другой.

Просмотр Activities можно включить, нажав на меню Activities или просто переместив мышь к левому верхнему краю экрана. Вид Activities делает рабочие пространства более интуитивными. Здесь есть предпросмотр всех открытых окон, а также запущенных приложений, во всех рабочих пространствах. Новые программы, документы и окна можно перетаскивать между рабочими пространствами. Также здесь можно закрывать несколько окон, не переключаясь в них. По умолчанию рабочих пространств два, но при необходимости их количество можно увеличить.

В вид Activities также интегрированы функции поиска и просмотра, которые ведут себя так же, как *Gnome Do*, и которые полезны, если нужная вам программа или файл не видны сразу.

3 **Apps-dash:** Как и все аспекты *Gnome Shell*, вид Applications предназначен для удобной работы с сенсорными экранами. Значки приложений могут запускать программы, а работающие приложения распознаются по свечению под названием программы. Щелчок по названию либо переведет вас в окно данной программы, запущенное последним, либо запустит новый экземпляр программы.

Еще одно замечательное нововведение – приборная панель в левой части экрана, где отображены запущенные программы; она также может служить для быстрого запуска избранных приложений.

4 **Уведомления:** Кроме этих функций, которые всегда на виду, в Gnome 3 есть более тонкие реформы дизайна интерфейса. Пример этого – новая система уведомлений, разработанная для связи между событиями без вмешательства пользователя. Уведомления возникают внизу экрана, чтобы не мешать удобству использования, что часто происходило в ранних релизах Gnome.

В новый рабочий стол Gnome также интегрированы сообщения на рабочий стол. Сейчас можно отвечать на сообщения IM и получать ветку разговора, не запуская программу и не переключаясь в нее из текущего окна.

Улучшение программ по умолчанию

В Gnome 3 не только рабочий стол капитально отремонтирован.

Несмотря на непривычный вид, Gnome 3 основан на том же наборе приложений, которые имелись в последних нескольких релизах. Эти программы были улучшены, отшлифованы и обновлены для нового инструментария *GTK 3*.

Файловый менеджер *Nautilus* сменил дизайн и обзавелся диалоговым окном для подключения к удаленным серверам. Новая панель инструментов стала опрятнее, строка состояния – умнее: она появляется только при обращении к ней; а управление в боковой панели организовано более интуитивно.

Проворный стандартный текстовый редактор Gnome, *Gedit,* содержит новое выскакивающее окно поиска. Сейчас он открывает и сохраняет сжатые файлы, и может похвастать более умной проверкой орфографии, способной теперь указывать даже на ошибки в комментариях в программном коде. Кроме этих визуальных изменений, произошли также переделки, невидимые глазу: например, документация *Gedit* переписана в формате Mallard. Формат Mallard – это язык гипертекстовой разметки, облегчающий написание документации разработчикам ПО.

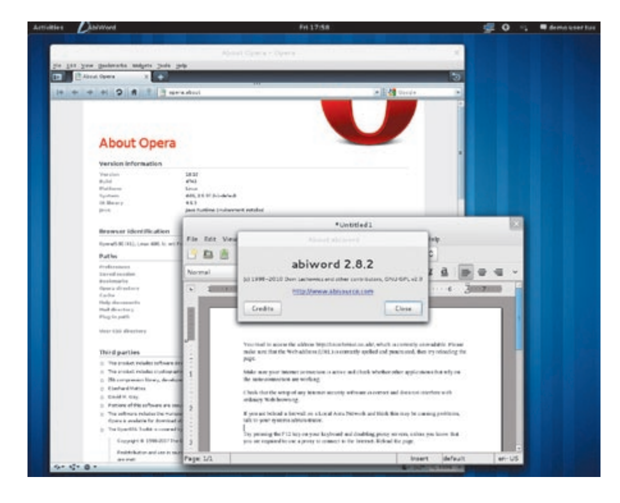
Просмотрщик документов *Evince* тоже использует Mallard для своей документации. *Evince* поддерживает несколько форматов документов, включая PDF, DJVU, TIFF, CBR, CBZ и так далее, а также формат XML Paper Specification (XPS).

Меню File и панель инструментов содержат новую кнопку Open Container Folder, по нажатию которой запускается *Nautilus* в папке, содержащей текущий просматриваемый документ.

В *Cheese,* программе создания снимков и видео с помощью web-камеры, переписан с нуля в Vala интерфейс пользователя. В *Cheese* также более 30 эффектов, и их предпросмотр работает вживую.

> «Разработчики могут перейти на Gnome 3 по своему желанию.»

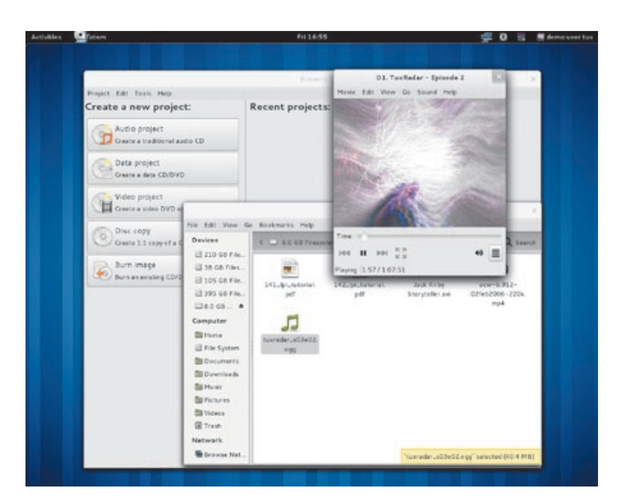
В web-браузере *Epiphany* есть несколько скрытых изменений, делающих его проворнее. Это и ненавязчивый менеджер скачиваний, и плавающая строка состояния, чтобы в браузере было больше места для web-страниц. Функция группировки вкладок по сайту, которая раньше предоставлялась расширением *Tab Groups*, сейчас встроена в сам браузер.



› **Приложения для Gnome 2 чувствуют себя в Gnome 3 как дома.**

Менеджер персональной информации *Evolution,* в котором собраны электронная почта, календарь, адресная книга и список задач, содержит новую опцию Picture Gallery в меню View. Наконец стала возможна распечатка Work View из календаря *Evolution* – необходимость в этом была заявлена еще в 2003.

Радикально усовершенствован клиент службы мгновенных сообщений *Empathy.* Он тесно интегрирован в новый рабочий стол Gnome и является его ключевым компонентом. Большая часть изменений в *Empathy* – полезные улучшения. Он запрашивает у вас пароль только при необходимости, а не предлагает вам сохранить пароль при создании учетной записи. Сейчас входящие звонки автоматически отклоняются, если вы на другом разговоре.



› ***Totem*, видеоплейер Gnome, основанный на *GStreamer*, сейчас содержит опции сохранения аудио- и видеопотоков.**

Gnome 2 и резервный режим

Хотя основные программы Gnome лучше встроены в новый рабочий стол Gnome 3, ваши любимые приложения Gnome также будут прекрасно работать. Если верить Шри Рамакришне [Sri Ramakrishna], редактору *Gnome Journal*, программа для Gnome 2 может существовать и в окружении Gnome 3. «Мы продолжаем поддерживать тот же API *GTK* от *GTK 2*, хотя они и устарели и программистам предлагается более новый API. Преимущество в том, что разработчики могут перейти на Gnome 3 по своему желанию», говорит он. Так как Gnome 3 – не просто визуальное обновление, но и дополнительные средства, такие как более гибкое положение, новые виджеты и так далее, можно ожидать, что в новый рабочий стол будут портированы еще больше приложений.

Gnome 3 требует аппаратного ускорения графики, что для некоторых может создать проблему. Если менеджер сеанса Gnome обнаружит, что возможностей графики недостаточно, рабочий стол автоматически переключится в резервный режим, где вместо *Gnome Shell* используются *Gnome Panel*, *Metacity* и *notification-daemon*, но они настроены так, что ведут себя как в *Gnome Shell*. »

FAQ –ЧаВо по Gnome 3

Выуживаем вопросы из вашего мозга на наши страницы.

Как протестировать Gnome 3?

Самый лучший способ попробовать Gnome 3 – через официальный live CD, размещенный на www.gnome3.org. Там есть и 32- и 64-разрядные версии для live CD, основанного на OpenSUSE. Можно записать ISO-образ live CD на пустую болванку и загрузиться с нее.

Другой способ – перенести образ на USB-брелок командой **dd.** Полагая, что USB-привод расположен в **/dev/sdc**, убедитесь, что он примонтирован, и выполните:

```
dd if=<path-to-ISO-image>Gnome_3.i686-0.1.0- Build1.1.iso
of=/dev/sdc
```

При загрузке образа CD в *VirtualBox* вы пока не сможете насладиться полнофункциональным рабочим столом Gnome 3, даже если у вас включено 3D-ускорение.

Будет ли мое любимое приложение для Gnome 2 работать в Gnome 3?

В Gnome 3 уже есть набор базовых программ. Вы сможете установить любую программу для Gnome 2 в Gnome 3. Если у вас есть свое приложение, основанное на *GTK*, лучше проверить его работу в тестовой среде, например Live CD.

Почему нет апплетов?

Во время мозгового штурма Gnome 3 разработчики и дизайнеры рассматривали базовые функции, которые обеспечиваются большей частью полезных апплетов, такие как часы, значок показа рабочего стола, и т. д. Поскольку все эти функции есть в *Gnome Shell*, она предназначена заменить этим все апплеты.

› В Gnome 3, разработчики хотят доставить рабочий стол, готовый к употреблению прямо из коробки.

Куда делось управление сворачиванием окон?

Отсутствие кнопок сворачивания (и распахивания) в заголовке окон в Gnome 3 изначально кажется ошибкой, но на самом деле это функция! Согласно команде дизайнеров, на рабочем столе Gnome 3 толку от этих кнопок мало, так как сворачивать окна некуда: нет ни дока, ни списка окон. В команде шли споры, не заменить ли кнопку сворачивания на кнопку, скрывающую окна, но в итоге решили не менять, поскольку обращение к месту, где содержатся скрытые окна, просто усложнит все дело.

Вместо этого, в новом рабочем столе, если вы хотите переключиться в определенное окно, выберите его из вида Activities. Окна можно растягивать, меняя их размер, а двойной щелчок сделает размер окна максимальным. Удаление этих двух кнопок также улучшило дружелюбную природу всего рабочего стола, так как теперь нельзя по ошибке распахнуть окно, когда вы хотите закрыть его.

То есть как, нет дока?

Как говорилось выше, устранение дока никак не повредит функциональности, и фактически это является основой не отвлекающей темы Gnome 3. Сейчас переключение между окнами происходит очень легко, и они не помешают той задаче, которая сейчас под рукой.

В качестве бонуса, удаление списка окон и дока сделало ненужной нижнюю панель. Ее отсутствие высвободило больше оперативного места всему рабочему столу, что будет роскошью для устройств с малым экраном, таких как нетбуки.

Смысл понятен, но почему опции настройки столь ограничены?

Одно из намерений дизайна Gnome 3 – свести к минимуму количество настроек и опций. Мнения в открытом сообществе по этому поводу разделились, так как многие опытные пользователи не любят, когда решения принимаются за них.

Control Center в Gnome 3 полностью сменил свой дизайн, включая содержимое всех панелей настройки. Одна из установок,

За кулисами

Кроме очевидных визуальных реформ, в Gnome 3 есть еще много изменений. Разработчики вернулись к кульманам. В интервью, данному *Gnome Journal*, разработчик Уильям Джон Мак-Канн рассказал, что одним из пунктов, который они желали исправить в Gnome 3, была чрезмерная модульность Gnome 2. «Gnome 2 стал напоминать Франкенштейна, собранного из кусочков, которые не всегда разрабатывались для четкой интеграции в рабочий стол».

Шри Рамакришна, редактор *Gnome Journal*, составил список улучшений в Gnome 3:

1 Число библиотек, требующихся для типичного приложения в Gnome 3, существенно снизилось, так как многие библиотеки GUI перемещены в *GTK*.

2 Движок отображения теперь Cairo, что дает разработчикам гибкость в отображении окон на устройствах разных типов, от смартфонов до LCD-мониторов и сенсорных экранов.

3 Темы *GTK* изменились, чтобы использовать синтаксис в стиле CSS, а не системы *Gtkrc*, которая использовалась ранее. Конечно, если дизайнеры работают с более знакомым синтаксисом, красота приложений только выиграет.

на которую наложено ограничение – выбор тем, хотя изменение фона рабочего стола все еще доступно из панели Background, где теперь можно брать изображения из Flickr.

Другой установкой по умолчанию, породившей множество дебатов, стала установка питания, которая приостанавливает работу машины при закрытии крышки ноутбука, а опция для изменения такого поведения отсутствует. На http://tinyurl.com/4gvlyt6 можно почитать более глубокие комментарии по этому поводу.

Но работа не останавливается. Сайт Gnome 3 упоминает, что «в будущем релизе планируются мощные расширения, которые предоставят пользователям и разработчикам захватывающие возможности настроек».

А где же GConf?

Система *GConf*, использовавшаяся для хранения опций настройки рабочего стола Gnome и программ, которые работают на его верхнем уровне, заменена на *GSettings*.

Если перейти с Gnome 2 на Gnome 3, что станет с моими настройками?

На встрече разработчиков *GSettings* в апреле 2010 разработчики пришли к соглашению о переносе настроек с *GConf* на *GSettings*.

Хакер *Gsettings* Маттиас Клазен [Matthias Clasen] объясняет, как это происходит. Сперва программы, которые заинтересованы в переносе настроек пользователя с *GConf* на *GSettings*, устанавливают файл преобразования – это файл ключей из каталога **/usr/share/GConf/gsettings**. Например:

```
[org.gnome.mousetweaks]
click-type-window-style=/desktop/gnome/accessibility/mouse/button_layout
animate-cursor=/desktop/gnome/accessibility/mouse/animate_cursor
```

Файл ключей требуется утилите *gsettings-data-convert*, которую устанавливает *GConf* и которая автоматически запускается при входе в систему. Утилита считывает файл и копирует данные из пользовательской базы данных *GConf* в *GSettings*. Затем она запоминает, какие преобразования уже производились в **~/.local/share/gsettings-data-convert**.

Такой процесс имеет ограничение в том, что может обрабатывать только ключи простых типов, такие как строки, целые, числа и числа с плавающей точкой, булевские значения и списки строк или целых чисел.

Маттиас Клазен объяснил, что во время разработки они пришли к выводу, что преобразовывать настройки пользователя в ключи, которые не раскрыты в пользовательском интерфейсе – плохая идея. Если продвинутым пользователям не понравится данная настройка и им захочется ее изменить, ничто не мешает им проделать это в командной строке.

Например, если вам хочется сменить тему, это можно выполнить с помощью утилиты командной строки *gsettings*:

```
gsettings set org.gnome.desktop.interface gtk-theme 'Clearlooks'
```

> «В будущем релизе планируются мощные расширения.»

Хм, то есть можно все еще использовать Gnome 2?

Конечно, можно. Ну, по крайней мере на данный момент.

Если верить разработчикам Gnome, некоторые части Gnome 2 поддерживать довольно сложно, и они не планируют продолжать поддержку некоторых частей ветки Gnome 2 после релиза Gnome 3. Тем не менее, рабочий стол Gnome 2 весьма стабилен, и он не исчезнет с зеркал вашего дистрибутива в ближайшем будущем.

Понятно… ну и как мне установить Gnome 3?

Компоненты Gnome 3 уже попали в тестируемые репозитории главных дистрибутивов. Когда вы держите в руках этот номер, релиз Gnome 3 уже осуществился. За подробными инструкциями по установке обратитесь на сайт Gnome 3 (www.gnome3.org). Добавьте подходящие репозитории, запустите менеджер пакетов вашего дистрибутива, сделайте себе мохито и наслаждайтесь блестящим новеньким рабочим столом!

Эволюция GTK 3

В основе Gnome 3 лежит новая библиотека *GTK 3*. Петер Маттис [Peter Matthis] взялся за нее в начале 1996, чтобы заменить закрытый инструментарий Motif, тогда применявшийся в *GIMP*. К середине 1996 года у него уже были первые работающие виджеты, которые включили в *GIMP* и назвали *GTK*, от *GIMP Toolkit*. К концу года добавили систему сигналов и переименовали инструментарий в *GTK+*.

В 1997 году в *GTK* резко прибавилось разработчиков, и позже в том году вышел первый релиз, независимый от *GIMP*. Вскоре уже несколько проектов, включая рабочий стол Gnome, использовали *GTK*.

Когда в 2002 году вышел *GTK 2.0*, он был уже совершенно зрелым и был широко принят, из-за чего разработчикам стало все труднее менять внутренности инструментария, чтобы API оставался стабильным.

Как говорит Тим Джаник [Tim Janik], куратор *GTK*, *GTK 3* был предложен на GTK+ Hackfest Berlin 2008. «Сообщества разработчиков, имеющих дело с *GTK*, а затем и Gnome, постепенно приняли эту идею и развили ее еще дальше, со списками функций и усилиями в разработке, возможными к реализации только на кодовой базе *GTK 3*, которым мы уже наслаждаемся».

Gnome 3 для разработчиков программ

Рабочий стол приносит пользу, только если в нем работают программы. Gnome 3 предлагает многое для облегчения жизни разработчиков.

Тим Ховард [Tim Howard], разработчик *F-Spot*, восхищен Gnome 3, в особенности его вводом GObject Introspection, который будет поддерживать программы, написанные на разных языках. «Эта C-библиотека дает разработчикам легкий способ поддерживать генерацию привязок [bindings] для разных языков в одном формате (GIR), и это должно стать огромным благом для пользователей, приведя к более быстрым обновлениям, большей функциональности и к еще лучшим программам», говорит он. Устранив необходимость вырабатывать привязки на уровне проекта и вместо этого положившись на общие привязки, Gnome 3 делает поддержку приложений проще.

Усовершенствование UI делается не только для пользователей. Согласно разработчику *Rhythmbox* Джонатану Мэттьюсу [Jonathan Matthews], способы взаимодействия с рабочим столом, которые Gnome 3 предлагает приложениям, определены более четко. «Это означает, что вам незачем тратить время, беспокоясь о том, делать ли панельный апплет, или имеет ли смысл сворачивание в системный лоток. Так как пользовательским темам и настройкам внешнего вида уделяется меньше внимания, можно быть уверенным, что конечный пользователь увидит все так, как вы и разрабатывали».

Чандни Верма [Chandni Verma], работающий над *Empathy*, считает, что самое большое облегчение – это стабилизация инструментария *GTK*. «Раньше [разработчики] тратили много времени на подгонку своих приложений для нового API, особенно когда какой-либо из классов перемещался или устаревал, или чтобы использовать преимущества новейших функций, в ущерб творческому подходу к функциям собственного приложения». LXF



Переверните страницу, чтобы подробнее рассмотреть GTK 3

GTK 3

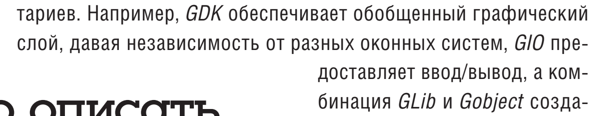
Пересборка GUI

Джон Ламби вооружился последней версией инструментария виджетов.

Слышите громовые раскаты надвигающейся революции? *GTK 3* — первая новая версия инструментария виджетов за последние девять лет, и она повлияет буквально на всех пользователей Linux, работающих в GUI. Если вам это кажется преувеличением, учтите, что на *GTK* основаны самые популярные рабочие столы Linux, такие как Gnome и *Xfce*, и *GTK* используется многими графическими приложениями из других рабочих столов, окружений окон и операционных систем. По сути, *GTK* можно описать как инструментарий для создания всего графического, работающего где угодно и на любом языке программирования.

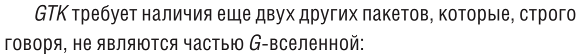
> «GTK можно описать как инструментарий всего графического.»

GTK – лишь один кирпич в пирамиде связанных инструментариев. Например, *GDK* обеспечивает обобщенный графический слой, давая независимость от разных оконных систем, *GIO* предоставляет ввод/вывод, а комбинация *GLib* и *Gobject* создает объектно-ориентированную основу с базовыми алгоритмами, утилиты для управления ими и т.д. А на вершине пирамиды расположен Gnome.



GTK требует наличия еще двух других пакетов, которые, строго говоря, не являются частью *G*-вселенной:

1. Cairo обеспечивает прорисовку графики.
2. *Pango* обеспечивает набор шрифтов.



При таком широком размахе *GTK* в нем неминуемы излишества. Чтобы создать в нем приложение, нужно знать не только *GTK*, но и многие другие кирпичи, лежащие ниже в пирамиде – и иногда портирование программы в другие окружения требует значительной переработки.

Цели GTK 3

Три года назад идеи об очистке и обновлении *GTK* были высказаны на Hackfest в Берлине, а около года назад появился и план действий [roadmap] – он приведен на www.tuxradar.com/content/gtk-3-plans-laid-out. Одной из самых главных задач было предоставление функций, в которых нуждался Gnome 3 (обратитесь к путеводителю по новому рабочему столу на стр. 24). Главными областями улучшений были:

GTK-приложения

GTK немного напоминает ядро Linux, только для графики (но он не жесткий, а, так сказать, «тягучий»); в нем самом многого не сделаешь – вся сила в программах, которые его применяют.

Итак, какие языки программирования и пакеты используют *GTK* сами по себе? Всех не перечислишь, но ниже вы найдете широко известные названия, которые стоит отметить:

1. Gnome 3, который подробно рассмотрен в этом номере.
2. *Metacity*, который в Gnome 3 был заменен на *Gnome Shell*, но все еще используется на старых машинах.
3. *Compiz* использует в *gtk-window-decorator* как Gnome, так и *Metacity*.
4. *Firefox* применяет *GTK* в ограниченных случаях, например, при настройках пользователя. *Firefox 4* все еще применяет *GTK 2*, но заметны признаки разработки внедрения туда *GTK 3*.
5. Также есть *PyGTK*, *Perl-GTK* и *JavaScript*, и другие интерпретируемые языки *GTK*.

1 Упрощение пирамиды.
2 Облегчение портирования приложений.
3 Помощь программистам и пользователям достичь желаемых эффектов, особенно с геометрией.
4 Добавление метода работы с устройствами ввода *XInput 2*.
5 Добавление нового способа работы с темами в API и пользовательских настроек на основе синтаксиса CSS.
6 Обогащение набора виджетов и введение расширений, например, новых базовых классов.
7 Оснащение аспектов интеграции с платформой, затрагиваемых при написании приложений.

GTK 3 не является конечным пунктом плана; уже идут разговоры о *GTK 4* (он-то выйдет раньше, чем через девять лет!) Если вы интересуетесь его эволюцией, то прогресс отслеживается на http://live.gnome.org/GTK+/Roadmap.

Столь радикальная переработка включает удаление обветшавших функций и изменение других, так что *GTK 3* собран так, чтобы сосуществовать с *GTK 2*, используя другие имена для библиотек. Это и хорошо, и плохо. Хорошо тем, что вы сможете установить и использовать его, не преобразовывая все ваши программы. Фактически библиотеки *GTK 3* уже есть в Fedora 14, замаскированные в пакете под названием **gtk3-2.90.5-1.fc14**. (Не тратьте свое время на испытания этого пакета; мы не смогли ни найти графическую программу, использующую его, ни собрать с ним что-либо). Fedora 15 будет содержать пакет уже «настоящего» *GTK 3* – возможно, это будет **gtk3-3.0.x** – и перейдет на Gnome 3, эксплуатирующий *GTK 3*. Ожидается, что в Fedora 15 и другие пакеты начнут использовать *GTK 3*, но не все. Последующие дистрибутивы, вероятно, будут вести себя так же.

В этом пункте у вас может возникнуть смутное подозрение насчет одной из отрицательных сторон таких решений: путаницы. Скажем, как узнать, есть ли у вас *GTK*? Это не так сложно: просто поищите в менеджере пакетов пакет с именем **gtk3****** или библиотеку *libgtk***3.****. Потом, как узнать, используется ли *GTK 3* в приложении или пакете *xyz*? Самый простой и легкий способ узнать это – если *xyz* решил поддерживать и *GTK 2*, и *GTK 3*, то в дистрибутиве будут два разных пакета *xyz* – каждый для своей версии. Есть надежда, что имена пакетов это отразят. Похоже, именно таким методом пользуются пакеты, желающие поддерживать *GTK 3*, не заставляя пользователей менять версию.



› Этот удобный переключатель GtkSwitch — новинка *GTK 3*.



› Ваши виджеты легко выстроить рядами и колонками.

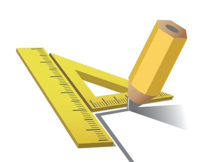
Это выводит нас на другой потенциальный недостаток сосуществования: дублирование. В Fedora 15 вовлечены следующие пакеты:

1 **PackageKit-gtk-module**
2 **ibus-gtk**
3 **libindicator**
4 **webkitgtk**
5 **spice-gtk**

Про те, что вне списка, сказать трудно. Попробуйте идентифицировать главный двоичный файл или файл библиотеки *xyz*, запустив для него команду:

```
ldd -v binary_filename
```

Если в выводе вы увидите ссылку на библиотеку *GTK*, она может дать ответ на ваш вопрос.



Итак, в отличие от типового обновления, простая установка пакета версии *GTK 3* на вашу машину либо не повлияет на все остальное, либо влияние будет незначительным. Фактическое же его подключение требует вмешательства разработчиков программ на его основе.

А надо ли вам заботиться об этом? Может, предоставить это поставщику дистрибутива? Да, возможно, но только если у вас нет чрезмерно перенастроенного внешнего вида *GTK* через личные настройки, или собственных *GTK*-приложений. Но если вы заметите изменения, это поможет вам разобраться в происходящем. Ясно, что раз вы создали собственные настройки или изменили системные, то понимали, что делаете.

Как насчет KDE?

Тем, кому хочется запускать *GTK*-программы в KDE и иметь возможность распространить на эти программы настройки и темы KDE, следует познакомиться с проектом Hugo: hhttp://hugo-kde.blogspot.com/2010/11/oxygen-gtk.html.

Особенности GTK 3

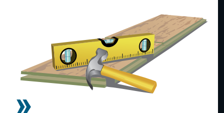
Рассмотрим подробнее некоторые из новых функций инструментария виджетов. В *GTK 3* есть система управления геометрией «height-for-width» и «width-for-height». Height-for-width значит, что виджет может менять размер требуемого места по вертикали за счет полученного места по горизонтали. Width-for-height работает сходным образом.

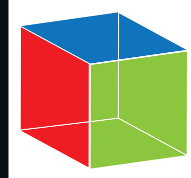
Пример работы такой системы – метка, которая «растекается» по ширине в контейнере, благодаря этому «оседая» по высоте. Это происходит полностью автоматически. Виджет должен решить, какое направление считается фиксированным, а какое – растягиваемым, то есть нужен ли height-for-width или width-for-height, а затем производит соответствующие вызовы. Обе функции запроса (на предпочтительную высоту или ширину) и вызовы расположения выполнятся рекурсивно вниз по потомкам виджета, если таковые есть, оценивая минимальный и натуральный размер каждого из них.

Расширения *XInput 2* обеспечивают поддержку более широкого класса функций и устройств ввода, таких как множественные устройства ввода, коды 32-разрядных клавиатур и расширенный интерфейс графических планшетов. Эта поддержка не обязательна при установке *GTK*, но по умолчанию включена. Приложения могут получать одновременный доступ к нескольким устройствам ввода и использовать добавочные «оси», предоставляемые устройствами, такие как сила нажатия и наклон. Координаты событий могут описывать позиции субпикселей, так как теперь их тип – число с плавающей точкой. (Такие интерфейсы – часть Устройств и Событий *GDK*, а не *GTK*.)

Введение в stylesheets

GTK 3 прорисовывает виджеты на основе стиля и темы. Конструкция **GtkStyle** из *GTK 2* уже устарела, как и спецификация стилей в пользовательском **.gtkrc-2.0**. Да и сам файл **.gtkrc-xx** тоже уста- »





рел. Их сменили StyleProvider API и стили типа CSS, задаваемые и настраиваемые пользователем. Имеется новый API для рендеринга и свежий API ThemeEngine. Это сильно улучшает способы, которыми программа и пользователь могут влиять на внешний вид виджета или класса виджетов. Также есть формальная поддержка разработки пользовательского движка тем документацией – хотя легче она от этого не стала.

Ниже перечислены свойства стиля, задаваемые в новом формате CSS:

1 **engine**
2 **background-color**
3 **color**
4 **border-color**
5 **font**
6 **margin**
7 **padding**
8 **background-image**
9 **border-width**
10 **border-radius**
11 **border-style**
12 **border-image**
13 **registered style properties**

Есть две разновидности зарегистрированных свойств стиля: характерные для движка стилевые свойства зарегистрированы в **GtkThemingEngines,** а характерные для виджета – в **Widgets.** Имена свойств имеют форму

```
-namespace-name
```

В этом примере **namespace** – имя движка темы или имя класса виджетов, соответственно, а **name** – имя свойства. Давайте-ка попробуем совместить это – вот пример CSS-подобного стиля в *GTK 3*:

```
GtkSwitch {
background-color: #122830;
color: #66c8fa;
font: Comic Sans 12;
-GtkSwitch-slider-width: 96;
}
GtkEntry {
background-color: #091929;
color: #99A9B9;
margin: 5 10 3 8;
-GtkEntry-inner-border: 10 3 8 5;
-GtkEntry-font: Times New Roman 10
}
```

В этом пункте следует отметить, что файл под названием **.gtkrc** или **.gtkrc-2.0** в домашнем каталоге пользователя ныне отсутствует. Этот файл содержал спецификации нескольких типов, а сейчас все они определяются иначе, а именно:

1 **Настройки, простые присваивания в формате name=value** Перешли в **settings.ini**.
2 **Стили** Перешли в стили CSS.
3 **Привязки клавиш** Новый синтаксис в формате CSS, того же типа, что новые стили.

Установки пользователя обычно управляются внешним менеджером окружения XSettings, который поставляется с рабочим столом, однако вы можете разместить их в новом файле **settings.ini**, расположение которого указано в экспортированной переменной окружения **XDG_CONFIG_HOME**. Если вы экспортируете **XDG_CONFIG_HOME** в **some_directory**,то *GTK 3* будет искать файл настроек в

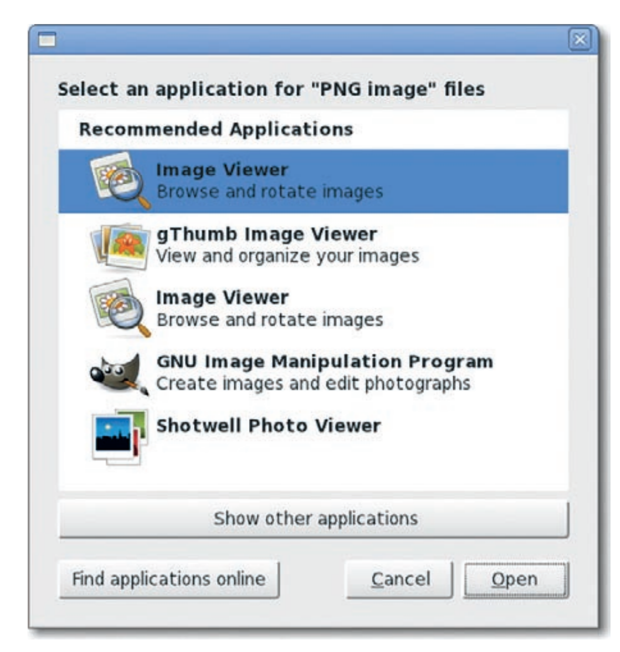
```
${XDG_CONFIG_HOME}/gtk-3.0/settings.ini
```

Можно видеть, что синтаксис настроек в нем немного отличается, но тем не менее до боли знаком:

```
export XDG_CONFIG_HOME=${HOME}/.xsettings
cat ${XDG_CONFIG_HOME}/gtk-3.0/settings.ini
[Settings]
gtk-theme-name=HighContrast
gtk-application-prefer-dark-theme=1
gtk-font-name=Times Roman 12
```



Точно – это ж воскресший INI-файл из MS-DOS! (Рано смеетесь: он мог бы стать и реестром). Формат name=value остался неизменным; новизна состоит в расположении файла и группировании разделов, область установок *GTK* называется **[Settings]**. В документации по *GTK 3* упоминается про файл.**gtkrc-3.0**, однако тестирование и обсуждение в списках рассылки указывают, что покамест *GTK 3* этот файл игнорирует – не утруждайте себя его созданием.



› Выберите приложение с помощью GTKAppChooser.

Свежие виджеты

Появилось несколько новых виджетов, и выполнен ряд усовершенствований для виджетов с прокруткой. Перечислим последние виджеты:

1 **GtkSwitch** Переключатель в легком стиле.
2 **GtkScale** Базовый класс для виджетов Horizonal и Vertical Scale.
3 **GtkCellArea** Абстрактный класс, пригодный для раскладки GtkCellRenderers.
4 **GtkCellAreaBox** Область ячейки, отображающая GtkCellRenderers в ряд или столбец.
5 **GtkCellAreaContext** Здесь хранится информация по геометрии рядов в GtkCellArea.
6 **GtkGrid** Распределяет виджеты по рядам и колонкам.
7 **GtkBox** Базовый класс для контейнеров.
8 **GtkButtonBox** Базовый класс для GtkHButtonBox и для GtkVButtonBox.
9 **GtkPaned** Другой базовый класс, для виджетов с двумя регулируемыми панелями.
10 **GtkSeparator** Базовый класс для GtkHSeparator и GtkVSeparator.
11 **GtkScrollbar** И еще один базовый класс – теперь для GtkHScrollbar и GtkVScrollbar.
12 **GtkScrollable** Интерфейс для виджетов с прокруткой.
13 **GtkAppChooser** Виджеты диалога и кнопки для выбора, которое из установленных приложений использовать для обработки конкретного файла.

Интеграция с платформой

Вообще говоря, тема платформенной интеграции охватывает все, что касается установки и использования программы в разных платформах и окружениях. Звучит несколько расплывчато, но для инструментария, которых охватывает столько окружений и должен подстраиваться под многочисленные и разнообразные подсистемы, лежащие в их основе, это действительно важно. Новые особенности *GTK 3* упрощают дело. А именно,

класс GtkApplication, реализованный как производный от класса GIO GApplication, обеспечивает средства связи окон с процессами и позволяет процессам отслеживать и контролировать друг друга через посредство сигналов и действий.

Термин «приложение» может породить некоторую путаницу – в нашем контексте он означает абстрактный объект, отличный от привычного определения выполняемой программы. Эту последнюю мы будем называть процессом. В частности, имя «приложения» не обязательно совпадает с именем исполняемого файла данного процесса.

«GTK 3 прорисовывает виджеты на основе стиля и темы.»

Например, некий процесс может идентифицироваться с определенным именем приложения и затем наблюдать за запуском/остановом другого процесса с таким же или другим именем приложения, и предпринимать действия, соответствующие ситуации. На **LXFDVD** помещен пример подобного случая. Процессы могут также обмениваться друг с другом различными простыми сигналами и более сложными объектно-ориентированными сообщениями, описывающими действия, которые следует выполнить.

В основе здесь лежит технология межпроцессного взаимодействия D-BUS (известная также как MessageBus), и она далеко не нова. Скажем, программный интерфейс для D-BUS уже много лет предоставляется в *GLib*. Например, вызовы метода, о которых мы упоминали выше, используют интерфейс D-BUS Introspect, который возвращает XML-строку, описывающую интерфейсы, методы и сигналы объекта. По сути, то, что произошло, представляет собой постепенную эволюцию и создание интерфейсов более высокого уровня в *GIO* и *GTK*. Данная эволюция не намерена останавливаться, и следующим шагом в ее развитии является новый GtkApplication.



Переход и сосуществование

Теперь займемся проблемами, сопутствующими переходу. Рассмотрим три типа пользователей: пользователь, который не настраивает и не разрабатывает приложения; пользователь, кто делает свои настройки, но не создает приложений; и пользователь, разрабатывающий ПО.

Первый будет приятно удивлен новым внешним видом и содержимым рабочего стола, без необходимости менять что-либо. Пользователь, склонный к настройке приложений, обнаружит, что Settings, Bindings и Styles, которые нужно было задать в **.gtkrc** или **.gtkrc-2.0**, видимо, куда-то подевались и (Bindings и Styles) переписаны в формате CSS, как описано выше. Действия третьего пользователя, как мы увидим, сильно зависят от природы его приложения и от языка программирования.

В языках семейства C многие функции были признаны устаревшими, а другие были заменены. Например, при рисовании вместо *GDK* применяются функции Cairo. Если программа пользователя требует любую из них, она должна быть преобразована, но, вообще говоря, в глобальном переписывании нужды нет. После этого, конечно, нужно пересобрать приложения, включив файлы и библиотеки *GTK 3*.

С интерпретируемыми языками ситуация менее ясна. В Python, знакомый пакет *PyGTK* устарел и заменен тандемом *PyGobject* и *Gobject-Introspection*. Без переписывания не обойтись, что может вызвать шок у обладателей больших библиотек приложений на *PyGTK*. В чем тут дело? *PyGTK* – пример привязки языка вручную («статической»). Разработчики годами считали необходимым доступ к внутренним переменным и приватным полям *GTK*, что все более усложняло поддержку, особенно при обновлении до *GTK 3*.

Gobject-Introspection предоставляет уровень, содержащий метаданные в библиотеке *Gobject*, где они могут быть доступны и из компилируемых, и из интерпретируемых языков, давая также способ полуавтоматического сканирования и распознавания синтаксиса языка программирования и сборки соответствующей библиотеки, которую можно привязать к языку. Для программ на Python это означает, что вместо импорта *PyGTK* они будут импортировать *GDK*, *GTK* и *Gobject* из репозитория *Gobject-Introspection*. Для языка Perl работа по переходу на *Gobject-Introspection* ведется на момент написания этой статьи.

И, наконец, следует переписать движки тем, ради реализации интерфейса *GtkThemingEngine* из *GTK 3*.

Попробуем

Как всегда, *Linux Format* шагает впереди (но не впереди паровоза) самых последних технологий; поэтому на **LXFDVD** помещено работающее окружение *GTK 3*, которое вы можете испробовать. Там содержится сборка *GTK 3* и сопутствующие пакеты, которые должны запускаться на любом дистрибутиве с *Glibc 2.11* или более новым, а также две небольших программки:

1 Программа на C и на Python, демонстрирующая переход с *GTK 2* на *GTK 3*. Под *GTK 2* это приложение использует виджет кнопки-переключателя, в сочетании с новым драйвером X-Telekinesis, позволяющим включать и выключать свет в редакции **LXF**. (Будьте социально ответственны, пожалуйста!)

2 Программа только на C, демонстрирующая технологию интеграции с платформой – процесс, следящий за стартом и остановкой другого процесса, основываясь на имени приложения.

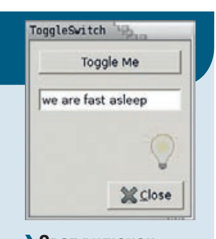
В версии программы *lightswitch* после перехода *togglebutton* заменен на новый виджет *GTK 3*, *lightswitch*. В случае с программой на C ее придется изменить и пересобрать с библиотеками *GTK 3*.

Для приложений Python это производится в два этапа. Начнем с исходного, из файла **togglebutton.py**:

1 Перейти с *PyGTK* на *PyGobject* с *Gobject_Introspection*, теперь в файле **togglegobj.py**.

2 Сменить виджет *togglebutton* на виджет *lightswitch* из *GTK 3*, сейчас файл **switchgobj.py**.

Инструкции по сборке и запуску программы прилагаются к пакету – все готово, чтобы вы могли попробовать самую свежую версию *GTK* и увидеть, что получается. LXF



› Свет включен.



› Свет выключен.



Тот самый герой GNU

Главный поборник интересов свободного ПО объясняет Майку Сондерсу, почему нам не стоит расслабляться...



Без Ричарда Мэтью Столлмена [Richard Matthew Stallman] не было бы и GNU, а без GNU не было бы дистрибутивов Linux, какими мы их знаем. РМС в 1983 году запустил проект GNU's Not Unix, чтобы создать совершенно новую операционную систему, а позднее — General Public License, для гарантии ее свободы. К 1991 году большая часть GNU была завершена, но ей не хватало ядра — и вот тут-то на сцену вышли Линус Торвальдс [Linus Torvalds] и его ядро Linux.

Несмотря на успех GNU/Linux, Столлмен не стремится к спокойной жизни: он неустанно участвует в кампаниях по защите наших софтверных свобод, предупреждая о потенциальных угрозах, вносимых новыми технологиями. Мы встретились с Ричардом в Институте Инжиниринга и Технологии [Institute of Engineering and Technology] в Лондоне, и передали ему некоторые ваши вопросы...

РМС: Прежде всего, я расскажу вам о свободном ПО – хочу, чтобы это было в интервью. Многие пользователи систем GNU/Linux и не слыхивали об идеях свободного ПО. Они не знают, что у нас есть идеи, что система существует благодаря этическим убеждениям, которые как-то выпали из представлений, ассоциируемых с понятием «открытый код».

Идея свободного ПО заключатся в том, что пользователи заслуживают свободы – в частности, контроля за своей системой. А проприетарные программы не дают пользователю контролировать свою систему. Проприетарные системы разделяют пользователей, заставляя их ощущать беспомощность. Разделяют, потому что пользователям запрещено распространять программы далее, а беспомощными они себя чувствуют, потому что не могут изменить программы, не имея доступа к исходному коду. Они не могут узнать, что этот код на самом деле выполняет.

Итак, проприетарная программа – это система несправедливой власти. Разработчик или владелец программы имеет власть над пользователями – несправедливую власть; а проприетарная программа – инструмент этой власти. Это несправедливо, и идея свободного ПО в том, чтобы избежать этой несправедливости и положить ей конец. Так что свободное ПО с уважением относится к свободе пользователя.

Итак, программа свободна, если она дает пользователю четыре основных свободы. Свобода номер ноль – это свобода работы с программами по своему желанию. Свобода номер один – свобода изучения исходного кода и возможности менять его в соответствии со своими нуждами. Свобода номер два – свобода помощи другим, свобода распространения точных копий по вашему желанию. Свобода номер три – свобода вклада в свое сообщество, то есть свобода распространения модифицированных вами копий.

И эти четыре свободы означают, что социальная система использования и распространения программ – это этическая система. При наличии этих четырех свобод пользователь контролирует программу. Без этих четырех свобод программа контролирует пользователя. С программами всегда так – либо одно, либо другое: в свободном ПО пользователь контролирует программу, в проприетарном ПО программа контролирует пользователя, а владелец контролирует программу, и через нее – пользователя.

Так что это не техническая проблема, а этическая. Это этическая проблема, которая возникает в результате использования определенной технологии. Но именно потому, что это этическая проблема, а не техническая, она важна – намного важнее любой чисто технической проблемы.

LXF: Большинство наших читателей — убежденные сторонники свободного ПО...

РМС: А что они понимают под «свободным»?

LXF: Ну, когда они с нами общаются, многие используют термин «свободное ПО» — а некоторые говорят «открытый код»...

РМС: А, видите, тут есть разница. Открытый код относится к другим идеям – к другой философии. А разница фундаментальная, она лежит на уровне ценностей. Это не разногласие в мелочах; это разногласие в основе. Наша цель – свободное общество, где пользователи обладают свободой.

Организаторы и лидеры открытого кода говорят, что их цель – лучшее качество кода. Разойтись дальше просто некуда: мы говорим о свободе и общественной солидарности, а они говорят о качестве.

LXF: Однако разве это не способ увести людей от одной системы и привести к другой?

РМС: Я вас не понимаю – это разные вещи.

LXF: Ну, допустим, ментальность компании, создающей проприетарное ПО, изменить в сторону GPL и свободного ПО непросто. А если свести к идее открытого кода через объяснение преимуществ его качества — идея будет воспринята и станет привычной, и можно будет ее расширить...

РМС: На самом деле – нельзя. Вот что я обнаружил: разговоры об открытом коде могут подвигнуть людей использовать некоторые свободные программы или иногда содействовать свободным программам. Но это укрепляет их ценности, а их-то нам и нужно изменить. Так что есть глубокая разница между тем, чтобы убедить человека использовать некую свободную программу или работать в свободной по большей части операционной системе, и тем, чтобы научить этого человека ценить свободу.

Это не одно и то же, и первое обычно не приводит ко второму. Фактически, когда широко распространяется философия открытого кода – что и произошло – она склонна отвращать умы от идей свободного ПО. И имеет тенденцию скрывать наше существование. Большинство статей о системе GNU даже не называют ее системой GNU или свободным ПО. Они описывают ее так, словно это открытый код, и создают впечатление, что мы – ее разработчики – согласны с идеями открытого кода, о которых читатели уже слышали и никогда не догадаются, за что мы на самом деле ратуем.

О СВОБОДЕ

«Владелец проприетарного ПО через программу контролирует и пользователя.»

LXF: То есть вы думаете, с оглядкой на начало 80-х, когда вы организовали проект GNU, что термин «свободное ПО» был наилучшим? Это я пытаюсь сказать «libre»... [«free» по-английски означает и «свободный», и «бесплатный», а французское слово «libre» значит «свободный», – *прим. пер.*]

РМС: Я тоже говорю «libre», по той же причине.

LXF: Услышав «свободное ПО», некоторые представляют себе мелкие шпионские программки в Windows.

РМС: Мне потребовалось время на осознание колоссальной важности этой разницы. В 1993 году, когда я сделал объявление [о GNU], я не отделял эти концепции. Прошло несколько лет, прежде, чем я их разделил. Итак, еще раз, в *Манифесте GNU*, появившемся в 1985 году, все еще смешиваются два значения слова «free». Тогда я и осознал необходимость подчеркнуть, что имеется в виду свободное в смысле свободы, а не бесплатное в смысле цены. Думайте о свободе слова, а не о бесплатном пиве.

Конечно, было бы лучше, осознай я это раньше. Хотя тогда я сказал бы в точности следующее: разночтения присутствуют по той причине, что в английском языке нет слова, однозначно выражающего именно то, что я хочу сказать. Единственное слово в значении свободный, то есть имеющий свободу – это «free», именно поэтому мы говорим «free/libre», потому что таким образом мы можем уточнить смысл.

Я заметил в вашем журнале [**LXF143**] утверждение о *LibreOffice*, которое является важной иллюстрацией. Sun приобрел *StarOffice* и выпустил



› Мы встретились с Ричардом во время его поездки по Европе. И заодно обнаружили, что у него страсть к вкусным сэндвичам.

его как свободное ПО под названием *OpenOffice.org*. Но люди из Sun, которые это сделали, не были сторонниками идей свободного ПО в политическом смысле. По сути, они были сторонниками открытого кода. И их целью было сделать их программу качественной и успешной – а не предоставить пользователям свободу.

Это их целью не являлось; хотя их код был свободным ПО и уважал свободу пользователя, но они об этом не думали в таком смысле. И они сделали список расширений, внеся в него проприетарные расширения. Примерно год назад, в мае, мы – Free Software Foundation [Фонд Свободного ПО] – объявили о нашем плане по созданию своего сайта расширений *OpenOffice.org*, где не будет несвободных расширений. Серьезной проблемой было то, что *OpenOffice.org* стал продвигать эти программы, давая тем самым понять, что несвободные программы – это нормально, ничего страшного.

И что я мог тут поделать? Мы предложили: давайте сделаем свой список расширений. *LibreOffice* использует наш список расширений – они его приняли от нас. Это решило проблему, а причина, по которой они это сделали, в том, что люди, создающие данную версию программы – активисты свободного ПО: им важна свобода. Они принимают решения ради свободы. Это показывает, что те, кто не думает о свободе или не ценит свободу, порой делают что-то не из желания содействовать делу свободы. Но на такое нельзя полагаться.

О ЦЕЛЯХ

«Иногда люди обнаруживают, что их мотивы — работать против свободы.»

Иногда люди обнаруживают, что их мотивам больше подходит работать против свободы. Изначально Линус Торвальдс в 1991 году разработал Linux как проприетарную программу. В 1992 году он выпустил ее под GNU GPL, и комбинация Linux с GNU позволила создать совершенно свободную операционную систему. Но он сделал это не потому, что ценил свободу: у него были другие мотивы, какие – я не знаю. И затем, в 1996 году, он стал включать в Linux несвободное ПО – двоичные куски для встроенных программ.

Узнав об этом, мы в FSF начали кампанию, чтобы хоть как-то противостоять – это было несколько лет назад. Мы стали подталкивать свободные дистрибутивы GNU/Linux к избавлению от этих кусков. А потом Александр Олива [Alexandre Oliva] начал распространять Linux-libre, то есть Linux без таких кусков.

LXF: Есть ли смысл в официальном дистрибутиве GNU? Кругом полно этих чисто свободных дистрибутивов GNU/Linux, таких, как Trisquel и gNewSense, и многие из них сдают позиции — это очень разрозненные проекты. Есть ли место для официального GNU? GNU/Linux от GNU?

PMC: Я полагаю, было бы неплохо, если бы такие проекты начали работать совместно. Но я не хочу затевать еще один дистрибутив, GNU – это выглядело бы пощечиной всем тем, кто сейчас работает над этими дистрибутивами, и я не хочу принимать чью-то сторону среди этих людей, отдавать кому-то предпочтение. Поступать так было бы довольно неудачно.

LXF: Многие из наших читателей хотят знать, чем именно вы занимаетесь, для примера. На вашем сайте есть фотографии, где вы работаете с ThinkPad, но сейчас вы его не рекомендуете.

PMC: Я не пользуюсь ThinkPad – это фотографии годичной давности. Сейчас у меня вот это устройство от Lemote – Yeeloong – можете называть его «remote», с китайским акцентом! [точнее, японским, – *прим. ред.*] Я выбрал это устройство, потому что оно свободно целиком, вплоть до BIOS. У него процессор типа MIPS, в китайской версии. В любом случае, суть в том, что он решает проблему.

LXF: Вам пришлось его модифицировать, или его можно купить как полностью свободное оборудование?

PMC: Я бы его так не назвал – я бы назвал его специализированным оборудованием. Но купить его можно.

LXF: И что у вас на нем работает?

PMC: Я работаю на gNewSense – это единственный совершенно свободный дистрибутив GNU/Linux, работающий в MIPS. Остальные – только для ПК, так что они не подходят. gNewSense также поддерживает ПК.

LXF: Возвращаясь к более глобальным проблемам: что бы вы назвали главной угрозой свободному ПО в 2011?

PMC: Их несколько. Юридические запреты, такие, например, как софтверные патенты в некоторых странах с неумной политикой. И еще есть законы, явно выступающие в качестве цензоров свободного ПО – например,

Digital Millennium Copyright Act в США, контролирующий свободное ПО, способное сломать цифровые наручники. В Европейском Союзе тоже есть подобные законы. И США, и ЕС пытаются протолкнуть свои мерзопакостные законы в другие страны, через договоры, которые этим странам предлагается подписать. Так что это – вредные правительства.

А еще есть препоны, чинимые производителями, частенько сотрудничающими с Microsoft. Например, есть масса примеров оборудования для ПК, работающего только с Windows. И, как правило, спецификации этого оборудования недоступны – безусловно неэтично продавать продукт, отказываясь сообщить, как он работает. Это недопустимо.

Еще одно препятствие – тенденция продавать компьютеры вместе с Windows. Я бы рекомендовал запретить и эту практику. И еще у некоторых компаний есть тенденция безвозмездно – ну или почти безвозмездно – передавать в дар школам свои программы, не относящиеся к свободному ПО. Это делают и Microsoft, и Apple – и я читал, что Фонд Гейтса [Gates Foundation] тоже этим занимается. Идея благотворительности Билла Гейтса – подсадить школьников на Windows и урвать еще больше денег. По-моему, это уже не благотворительность.

LXF: Один из наших читателей задал вопрос: возможен ли мир, где есть только свободное ПО? Это по-прежнему является главной целью?

РМС: Да, цель именно такова, полагаю. Это моя цель. Может быть, совсем избавиться от остатков несвободного ПО невозможно. Ведь в конце концов, за 200 лет аболиционизма мы так и не избавились от рабства. Есть места, где люди, по сути, рабы. Я читал, что иностранные рабочие в Великобритании по сути являются рабами, поскольку если они вздумают жаловаться, их просто депортируют.

Так что полностью искоренить все формы злоупотребления весьма затруднительно, но я уверен, что общество, где проприетарное ПО является исключением, вполне возможно, если мы все вместе потребуем такого общества.

LXF: Намного больше людей сейчас используют смартфоны и планшеты в качестве компьютерных платформ, со всеми их магазинами приложений...

РМС: На самом деле, это ничего не меняет. Смартфон – это компьютер: он не создан с помощью компьютера; действия, которые он выполняет – это действия компьютера. Так что все, что мы говорим о компьютерах – что программы, с которыми вы работаете, должны быть свободными, и вам надо настаивать на этом – применимо в равной мере и к смартфонам. И к планшетам тоже.

Что сказать по поводу этих магазинов приложений? Ну, во-первых, магазины приложений Apple и Microsoft запрещают свободное ПО и разрешают только несвободные программы. И это доказывает их вредность. Но помните, что их база – несвободная ОС. Желая жить свободно, надо требовать не приложений, относящихся к свободному ПО, а свободной ОС. Так что и «iMoan [moan – *англ.* стенание]», и «iBad [bad – *англ.* плохой]» [обыгрывается созвучие с iPhone и «iPad», – *прим. пер.*] по сути своей зло.

Они вас и близко к свободе не приведут, так что не стоит ими пользоваться. И Windows Phoney 7 [снова игра слов: phone – телефон, phoney – обман, – *прим. ред.*] не собирается давать вам свободу, так что эти продукты, очевидно, абсолютно вредны. Они еще усугубляют зло, распространяя несвободные приложения, и только несвободные приложения; но и без этого они было бы неприемлемы.

А Android – иной случай. Исходный код Android – свободный, в том виде, как его выпустил Google, но они используют лицензию, не являющуюся копилефтом, кроме Linux, который идет под GPL v2. И в результате лицензия не защищает пользователя от запертости или Тивоизации [Tivoization – практика создания аппаратно-программных систем, у которых ПО имеет копилефт-лицензию, но аппаратура не дает запускать модифицированную версию ПО, – *прим. пер.*] – то есть от практики превращения свободной программы в несвободную, запрещающую пользователю установить и использовать собственную версию. Очень многие разновидности смартфонов с Android блокируют установку иной версии программ.

О MICROSOFT

«Благотворительность Гейтса – подсадив школьников на Windows, урвать денег.»

LXF: Продолжается битва за умы большого количества людей – они даже не знают, что такое исходный код.

РМС: Совершенно верно.

LXF: Например, мои родители – это как раз случай, когда надо попытаться найти правильный подход...

РМС: Я использую аналогию с рецептами. Это хорошая аналогия, потому что программа во многом похожа на рецепт. И то, и другое – определенная последовательность действий для получения желаемого результата. И если вы посмотрите, как повара используют рецепты, то увидите, что на практике они пользуются теми же четырьмя свободами в применении рецептов. Повара готовят по рецептам свободно, они изучают и изменяют их, когда хотят, они распространяют копии; и если они вносят свои изменения, то могут поделиться измененной версией.

Итак, представьте себе, что бизнесмены и государство решили навязать проприетарные рецепты. Предположим, государство заявило: с завтрашнего дня, если вы скопируете или измените свой рецепт, мы посадим вас в тюрьму и объявим пиратом. Представьте себе, как обозлятся повара. Многие из тех, кто ничего не знает о программировании, поймут это. Государство не пыталось сделать так с рецептами – зато попыталось сделать с программами.

LXF: И напоследок: вы все еще балуетесь с компьютером?

РМС: Время от времени я хакерствую, но не в программировании. Пожалуй, Guantanamero (http://stallman.org/guantanamero.html) было хакерством, пародией на песню Guantanamera. С тех пор я написал еще одну пародию на испанском.

LXF: Будет ли продолжение у Песни свободного ПО?

РМС: Думаю, нет! LXF

› *Emacs*, GPL, *GCC* и многим другим мы обязаны Столлмену.



17 перемен к лучшему в именах

Что в имени тебе моем? Когда речь идет о свободном ПО, имена –это сокращения, иногда шутливые, и множество заглавных букв. **Нейл Ботвик** рассматривает те, которым особо не повезло с выбором.

01 Mageia

Mageia – возможно, самое неудачное название дистрибутива среди всех, которые еще не стерлись из памяти. Сперва мы думали, что это 'Magela', потому что прочли объявление о релизе, напечатанное шрифтом без засечек. Но наша версия была хотя бы произносимой, в отличие от изначального имени. Когда на форумах проекта больше сообщений о произношении его названия, чем об исходном коде, значит, название действительно неудачное. К счастью, этот проект – ответвление от Mandriva, урожденной Mandrake, и разработчикам не привыкать к смене имен по ходу работы.

> «Кроме написания и произношения, проблем у имени нет.»

02 GNU/Linux

При всем уважении к РМС, имя GNU/Linux ужасно – оно плохо выглядит, а звучит еще хуже. Как вы это произносите? «Гну дробь Линукс»? Вряд ли. 'Gnu Linux'? Тоже нет, потому что это озвучивается как 'New Linux' и провоцирует вопрос: а что же «не так» со старым Linux? Не то что мы дискредитируем работу, проделанную GNU, но имя проекту они выбрали неуклюжее.

› **Произнесите-ка подряд три гласных. Неважно, хорош ли окажется дистрибутив Mageia — неудачное имя помешает ему добиться успеха.**

03 Имена прилипают

Как бы вы ни назвали свой проект, стоит людям начать им пользоваться, имя тут же прилипнет. Привычные имена менять сложно – кто ж назовет «Марафоном» батончики «Сникерс»? Выбирайте имя тщательно: оно, возможно, переживет даже вашу заинтересованность в самом проекте.

04 Gnome

Как это произносить? Годы обучения (некоторым из нас случалось посещать занятия) научили нас не произносить 'G', но многие люди его озвучивают. Путаница усугубляется тем, что исходный акроним GNOME становится все менее понятным, и многие теперь называют эту среду Gnome. Так что, кроме написания и произношения, проблем у этого имени нет. Как удачно для проекта, который урезает опции настройки, чтобы создавать меньше путаницы для пользователей!

05 KOffice

KOffice – хорошее имя: очевидно, что это офисный пакет для KDE. Имена компонентов тоже имеют смысл: *KWord*, *KSpread*, *KPresenter*, *KChart*, *Krita*. Так, а откуда взялось последнее имя? Разве тот, кто отвечал за выбор узнаваемых имен, ушел в отпуск? Хотя для большинства программ семейства и использовались осмысленные имена, некоторые члены команды *KOffice* сочли нужным изменить устоявшееся название программы и преобразовали его в *Calligra* – даже не 'Kalligra'.

06 Lignux

Если вам не нравится имя GNU/Linux, то что скажете насчет Lignux? Вот уж действительно. Это было одно из названий Linux, предложенное приверженцами GNU. Встраивание их рекурсивного акронима в слово Linux выглядит еще безобразнее, чем приклеивание его в качестве префикса. Слава богу, что эта идея была отвергнута.

07 *buntu

В именах Ubuntu или даже Kubuntu/Lubuntu нет ничего плохого. Ubuntu – неплохое имя, простое для произношения и мгновенно узнаваемое. Проблема только в том, что каждая следующая переработка Ubuntu получает имя «нечто-buntu». Это не просто раздражает отсутствием воображения, но и мешает дистрибутиву переключиться на другую базу или стать

самостоятельным, узнаваемым дистрибутивом, как Linux Mint, потому что имя всегда будет говорить о том, что это – очередная переработка Ubuntu. А дальше что – Mikebuntu?

08 gNewSense

Бездумный выбор имен без учета произношения вышел на новый уровень в 2006 году с выходом дистрибутива gNewSense. Какой бы ни была идея имени, практика непроизносимого 'g' привела к малоприятному названию [NewSense звучит похоже на nuisance – досадная помеха, – *прим. пер.*]. Возможно, однажды мы увидим дистрибутив, специализирующий на коммерческом ПО с названием gNonSense [**Non**-Sense = nonsense – чушь, – *прим. пер.*].

09 Рекурсивные акронимы

Все это началось с Gnu's Not Unix, и PHP Hypertext Processor был разумной реализацией. Но сейчас рекурсивные акронимы превратились в проклятие. На SourceForge тьма проектов с именами в виде изощренных рекурсивных акронимов. Впечатляют они лишь немногих, а еще меньше людей вообще проявляют к ним интерес.

> «Рекурсивные акронимы превратились в проклятие.»

10 Debian

Debian показывает, что можно отступить от правил и все же получить имя, которое работает хорошо. Для большинства пользователей это сокращение – пустой звук, но оно составлено из имен основателя проекта и его подруги и превратилось в самостоятельное узнаваемое слово. Название выглядит естественно, и его произношение вполне удобно – хотя вряд ли именно к этому стремился Иэн Мэрдок [Ian Murdock] 18 лет тому назад.

11 Кчтопопало

Мы не видим ничего дурного в том, чтобы имена программ KDE начинались на K, а программы Gnome – на G, но иногда это смотрится как притянутое за уши. Название *K3b* говорит нам о том, что это – программа KDE, но и все. Неплохо было бы сообщить еще и о том, что программа делает.

12 Linux

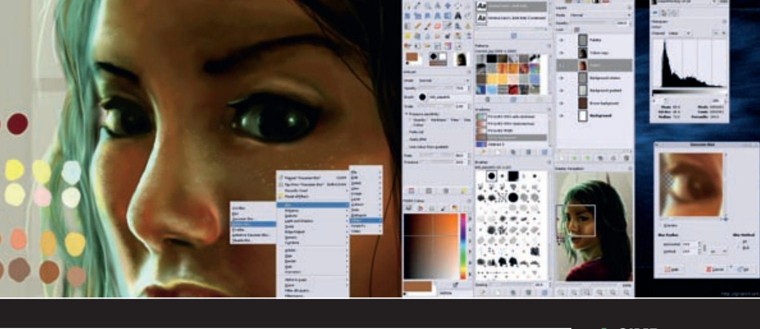
Слово Linux тоже не лишено проблем с озвучиванием, хотя уважение к Линусу диктует только один вариант произношения его имени. Но надо ли, чтобы проекты назывались по именам их основателей, или все же не надо?

13 Эпонимные проекты

Идея назвать проект своим именем поначалу кажется лестной. Но вдруг проект вам надоест и вы скинете его на других? Это не лучше, чем назвать детей в честь самих себя. Вот Билл Гейтс [William H. Gates III] так не сделал – ни со своими детьми, ни со своей компанией, ни со своими разработками... в отличие от Майка Сондерса I.

14 Micro$oft

Только ли мы одни кривимся, увидев издевательские сокращения типа M$ применительно к Windows и Microsoft? Разве вы настолько не уверены в превосходстве любимой ОС, что опускаетесь до детских дразнилок? Поступать так на форуме Linux бессмысленно, а на Windows-форумах или в смешанной пользовательской среде – грубо и глупо.



› *GIMP*: программа нравится, но имя слегка неловкое...

15 GIMP

Классический пример ключевой программы Open Source с неподходящим именем, менять которое уже слишком поздно. Сокращение до *GIMP* для GNU Image Manipulation Program – GNU-программы манипуляций с изображениями – могло вначале неплохо служить разрабочикам, но теперь не говорить же пользователям «я отгимпировал изображение» [gimp – *англ.* хромой]. А в результате пользователи *GIMP* выражаются «отфотошопили», во избежание непонимания обычных людей.

16 Windows ничуть не лучше

Пользователи Windows часто приводят бессмысленные имена как аргумент в спорах, указывающий на слабость Linux, противопоставляя им *Internet Explorer* или *Word*, но совершенно забывая про *Excel*, *Outlook* и прочее. Когда им на это указывают, они смущаются, но набрасываются на вас, если привести им в качестве аргумента не менее бессмысленное название *Acrobat*. «Это сторонняя программа, она не считается», кричат они, хотя вы можете отпарировать, что в Linux все программы сторонние!

17 Choqok

Если кто не знает, это название отличного клиента для ведения микроблогов по-древнеперсидски значит «воробей». А если вас не устрашает древнеперсидский, задумайтесь об отношении воробья к микроблог-клиентам. Все же должно быть больше связи, чем просто факт, что воробьи, как и многие другие птицы, чирикают. LXF

Хорошие новости



Не забудем и о хороших, осмысленных и удобопроизносимых именах, не вызывающих случайных и глупых ассоциаций – в том числе *AbiWord*, *KMail*, *MythTV*, *GnuCash*, *FlightGear*, *Digikam*, *Mplayer*, *Audacity*, *VirtualBox* и многие другие. Есть и такие имена, которые не имеют прямой связи с функцией приложения, но стали нарицательными благодаря усилиям, вложенным разработчиками в успех проекта – например, *Firefox*.

Организуйте жизнь с Tomboy

Джоно Бэкон собирает воедино беспорядочные заметки, записки и идеи с помощью Tomboy, попутно объясняя, как это делается.



Мы живем в море информации. Раньше компьютеры применялись для автономной обработки данных, но теперь, при наличии электронной почты, социальных сетей и распределенной обработки данных и терпимости к одновременной работе над множеством задач, нам грозит наводнение информационных потоков.

Есть множество инструментов, по уверениям разработчиков призванных помочь нам в этом: программы для планирования времени, управления проектами, составления списков текущих дел... список бесконечен. Возможно, вы думаете, что *Tomboy* – просто еще одна программа для ведения заметок; что тут особенного? Но изюминка этого приложения – в легкости внедрения его в жизнь. Именно простота, в сочетании с рядом организационных приемов, и есть то, чем мы хотим поделиться с вами. Сначала рассмотрим базовые приемы работы с программой, а затем изучим методы приспособления *Tomboy* к вашим потребностям.



В большинстве дистрибутивов *Tomboy* устанавливается через менеджер пакетов, или можно скачать программу с http://projects.gnome.org/tomboy. Установив пакет *Tomboy*, запустите его, и вы увидите окно, видом похожее на рисунок справа, хотя и с меньшим количеством заметок. Для начала создадим заметку. Выберите команды File > New [Файл > Создать], и появится новое окно. Оно выглядит и ведет себя, как окно текстового редактора. Новая заметка всегда содержит заголовок (он подчеркнут) и типовой текст. Щелкните мышью по заголовку и впишите текст, например: ‘Читать Linux Format’, и теперь, вернувшись в главное окно, отображающее список ваших заметок, вы увидите в нем и свою новую заметку.

Теперь к заметке, снабженной заголовком, можно добавлять информацию. Каждая заметка *Tomboy* имеет опции форматирования, позволяющие ее оживить. Например, вставим маркеры списка. Добавим немного текста (например ‘Купить последний выпуск’). Теперь выделите текст мышью, щелкните по кнопке Text [Текст] и выберите опцию Bullets [Маркеры]. Далее действуйте так, как в обычном текстовом процессоре: в конце каждой строки нажимайте клавишу Enter, чтобы добавить новую строку маркированного списка. Просмотрев опции форматирования, вы увидите, что они позволяют выделять текст полужирным шрифтом, курсивом, изменять размер шрифта, выделять фрагменты и т. д. Поэкспериментируйте с различными опциями, чтобы разобраться с ними.

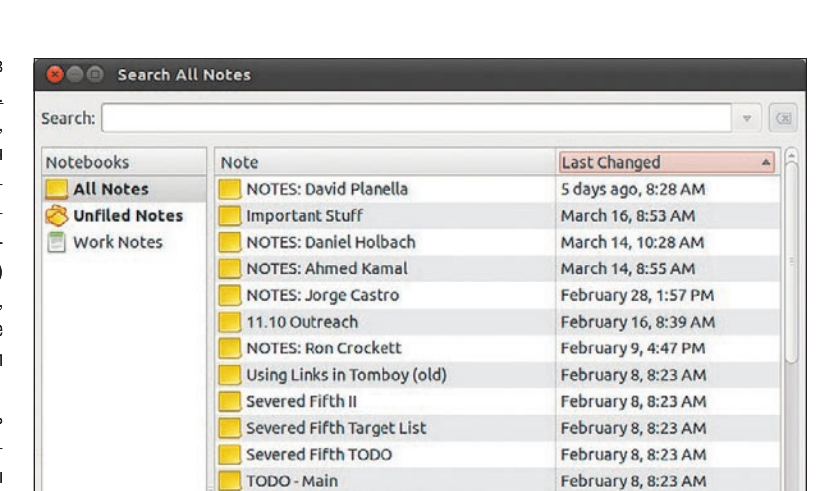
Сохранение плана дня

Важная и остроумная особенность *Tomboy* – отсутствие необходимости в сохранении. Каждый раз, завершив редактирование заметки, просто закройте окно, и все добавленное вами сохранится автоматически. Благодаря этому пользоваться *Tomboy* – сплошное удовольствие; не нужно ни думать о том, где находятся ваши заметки, ни беспокоиться о том, что вы потеряете важную информацию, потому что забыли ее сохранить.

Пока *Tomboy* не выглядит чем-то большим, чем продвинутый текстовый редактор. Все, что мы проделали, можно было бы выполнить в *Gedit* или аналогичной программе. Но *Tomboy* обретает крылья благодаря своей поддержке ссылок между заметками. Продемонстрируем это: перейдите в окно со списком заметок и добавьте заметку с названием Журналы, а в текст добавьте названия журналов, которые вы хотели бы прочесть. Теперь перейдите к первой заметке (‘Читать Linux Format’). В ее тело добавьте некоторый текст, например: ‘Читать другие Журналы’. Когда вы это сделаете, вы увидите, что слово Журналы автоматически выделится как гиперссылка. Пройдя по этой ссылке, вы откроете вашу вторую заметку, Журналы.

> «Tomboy обретает крылья благодаря своей поддержке ссылок.»

Принцип работы этой функции состоит в том, что при упоминании названия заметки внутри другой заметки, *Tomboy* автоматически добавляет ссылку. Ссылки нечувствительны к регистру символов, поэтому на заметку Журналы можно ссылаться как на ‘журналы’. Внутри тела заметки, в любом тексте, с любым форматированием и любым стилем, можно ссылаться на имена других заметок, превращая их в ссылки.



› **Окно со списком заметок *Tomboy*.**

Это преобразование в ссылки очень удобно – оно делает ваши заметки крайне полезными и контекстно-связанными. Если у вас есть заметка с названием To-do, то при каждом ее упоминании в другой заметке вы автоматически получаете возможность просмотра списка ваших задач в контексте. С учетом этого вы можете присваивать своим заметкам осмысленные имена и автоматически связывать их между собой, где это уместно.

Вы, вероятно, очень быстро обрастете массой заметок, создавая, например, списки покупок, списки задач, заметки с идеями, намеченными к реализации, а также важными сведениями, которые нужно запомнить. Некоторые из заметок могут стать для вас родом стандартной мебели – это те, на которые часто ссылаются и обновляют для отслеживания своей повседневной жизни. Некоторые заметки будут временными (например, такие, которые используются для управления конкретным проектом или задачей).

Даже при изобилии заметок и информации вы сможете быстро и легко находить нужные сведения. К счастью, главное окно со списком заметок имеет мощную встроенную функцию поиска не только по названиям заметок, но и по их содержимому. Это значит, что каждое слово и каждый фрагмент информации в заметках индексируются и доступны для поиска в главном окне приложения.

Помимо простоты и элегантности *Tomboy* как приложения для ведения заметок, оно является и ключевой частью рабочего стола Gnome, а также поставляется в составе некоторых дистрибутивов (например, Ubuntu). Это дает удобные возможности интеграции с рабочим столом. В зависимости от вашего дистрибутива, одни функции интеграции с рабочим столом будут активированы по умолчанию, а другие потребуется включить вручную. *Tomboy* поддерживает ряд сторонних дополнений, доступных через диалоговое окно Edit > Preferences [Правка > Параметры].

Добавочная стоимость

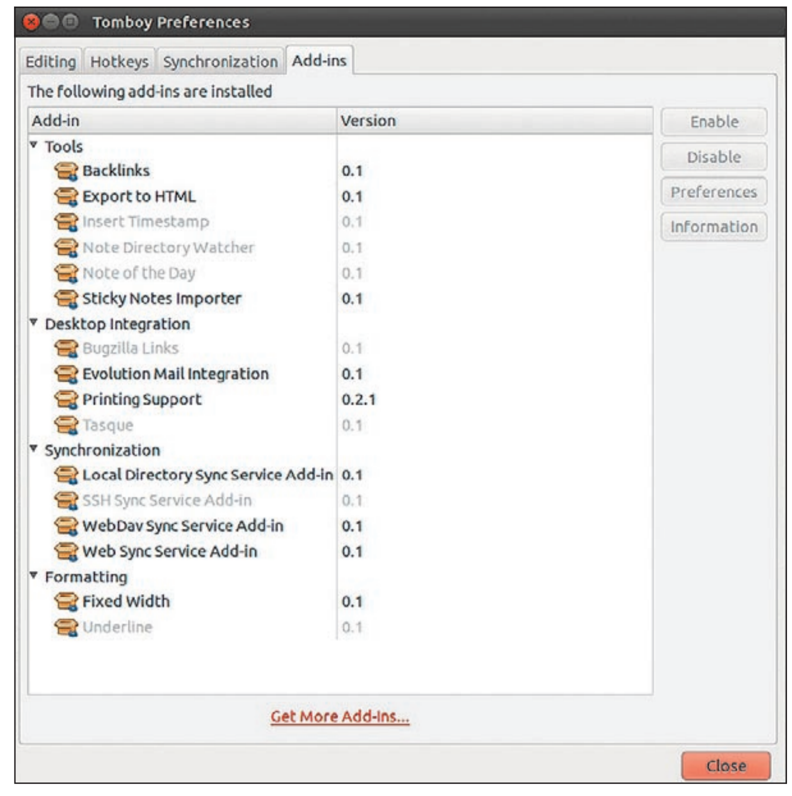
Одно из первых дополнений, с которым следует познакомиться – поддержка уведомлений и индикаторов. Если у вас Ubuntu с полной системой уведомлений и индикаторов, то индикатор *Tomboy* станет доступен при выборе соответствующего дополнения. На панели индикаторов появится небольшой значок в виде заметки – он дает быстрый »



Использование блокнотов

Еще один совет по использованию *Tomboy* касается использования блокнотов. Блокноты – это подборки взаимосвязанных заметок (например, у вас может быть рабочий блокнот и блокнот для личных заметок). Чтобы добавить блокнот, перейдите в главное окно *Tomboy* со списком всех заметок, щелкните правой кнопкой мыши по панели Notebooks [Блокноты] и выберите команду New Notebook [Создать блокнот...].





› *Tomboy* легко расширяется множеством сторонних дополнений.

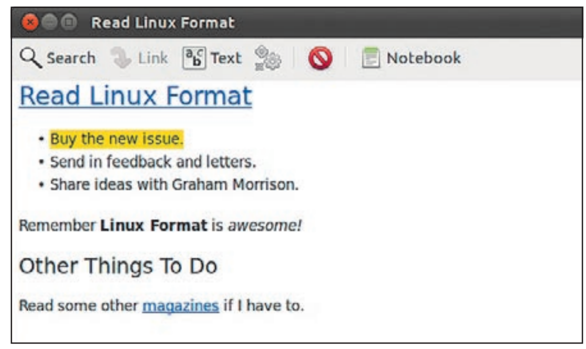
доступ к основным функциям *Tomboy*, а также к недавно просмотренным заметкам. Если у вас дистрибутив с более старомодной областью уведомлений, то и в этом случае будет доступен дополнительный значок уведомления.

Еще одна удобная функция – связь *Tomboy* с почтовым клиентом *Evolution* и клиентом службы календарей. Установив дополнение для поддержки *Evolution*, вы сможете перетаскивать почтовые сообщения или цепочки дискуссий из *Evolution* в заметки *Tomboy*, и к ссылкам добавится значок в виде почтового конверта. При щелчке мышью по такой ссылке почтовое сообщение загрузится в *Evolution*.

На данном этапе не худо бы вспомнить, что *Tomboy* – всего лишь инструмент: он не сделает вас более организованным, если вы сами не стремитесь к самоорганизации. Теперь, получив общее представление о базовых функциях, сконцентрируемся на различных методах, позволяющих добиться, чтобы программа *Tomboy* заняла центральное место в вашей работе и способах обработки важной для вас информации.

Первый шаг – тщательное обдумывание вашей работы, ваших хобби и различных способов, которыми можно сделать *Tomboy* полезным.





› Заметки *Tomboy* предоставляют множество различных опций форматирования, включая знакомые полужирный шрифт и курсив.

Для этого разделите свою жизнь на две основных области: профессиональную и личную. Рассмотрим задачи, решаемые в ходе вашей профессиональной деятельности. Делаете ли вы ежедневно произвольные заметки о делах, которые необходимо выполнить? Создаете ли вы заметки при ведении дискуссий? Часто ли вам требуется искать общую информацию, например, номера для телефонных конференций? Заметки *Tomboy* хорошо обслуживают каждую из этих областей. Создайте типовые заметки, которые вы будете использовать регулярно. Например, можно создать заметку Важное, содержащую общую информацию, на которую вы часто ссылаетесь, а затем – заметку Сегодня, которая послужит общим шаблоном для всех заметок о делах, запланированных на конкретный день.

Теперь подумаем о личной жизни. Часто ли вам приходится искать одну и ту же информацию (например, контактные телефонные номера вашего банка, дантиста, парикмахера и т.д.)? Составляете ли вы списки вещей для покупки при очередном походе по магазинам? Заметки *Tomboy* могут быть отличным подспорьем для записи идей насчет подарков вашим родным, друзьям и знакомым на праздники, дни рождения и прочие подобные случаи. Просто продумайте все свои способы работы с информацией.

Управление процессом работы

В профессиональной деятельности подавляющее большинство из нас сталкиваются со следующими двумя вещами, которые имеют место всегда:

1 **Задачи** То, что следует выполнить. Они варьируются от самых мелких (например, вынос мусора) до более крупных и сложных (например, работа над годовым отчетом).

2 **Проекты** Более масштабные цели, состоящие из серий сгруппированных задач.

Tomboy – замечательное средство автоматизации рабочего процесса [workflow]. Для каждого проекта можно создать заметку *Tomboy*, чтобы отслеживать этапы его выполнения. Заметки могут включать такие разделы:

1 **Проект [Project]** Название проекта.

2 **Цель [Goal]** Простое предложение, кратко резюмирующее результат проекта.

3 **Критерии успеха [Success criteria]** Описание результата, который считается успешным. Важно понимать, что на любой стадии проекта вы должны иметь возможность задать вопрос об успешности полученного результата с ответом ‘да’ или ‘нет’. Если вы получаете ответ ‘возможно’, критерии успеха надо уточнить.

4 **Участники [People involved]** Список всех участников проекта и порученные им разделы проекта.

5 **Сроки [Deadlines]** – согласованный список предельных сроков завершения проекта и его ключевых этапов.

6 **Заметки [Notes]** Создайте внутри вашей заметки общий раздел для занесения в него произвольных сведений, которые вы не хотите забыть.

Ваши заметки повсюду

Хранить заметки на своем компьютере удобно, но еще удобнее их распределенное хранение. Вот несколько опций по хранению заметок, привлечению к ним должного внимания и получению к ним доступа во время разъездов:

» **Ubuntu One** Этот сервис (http://one.ubuntu.com) позволяет синхронизирует заметки между компьютерами Ubuntu и Windows, а также получать к ним доступ через web-сервис Ubuntu One.

» **Tomdroid** Ищите это приложение в разделе Productivity на Android Market (https://market.android.com). Если у вас есть смартфон Android и вы хотите получать с него доступ к заметкам, установите *Tomdroid*.

» **Snowy** – если вы предпочитаете настроить свой web-интерфейс для синхронизации заметок, обратите внимание на сервер *Snowy* (https://live.gnome.org/Snowy), весьма полезный для корпоративных развертываний *Tomboy*.



За регулярным обновлением заметки проекта нужно следить. При изменении сроков завершения происходят рабочие согласования, появляется новая информация или происходят другие события; обязательно отражайте это в заметке проекта.

В ходе работы над проектом вам, скорее всего, потребуется делать регулярные телефонные звонки или проводить встречи с другими участниками. Рассмотрим полезный прием выполнения этой работы, позволяющий структурировать расписание встреч и гарантировать, что и вы, и тот участник проекта, с которым вы поддерживаете связь, всегда будете иметь список текущих задач. Сначала создайте заметку для каждого своего сотрудника – например, для Джо Блоггза, как одного из участников, названных в заметке проекта – и обеспечьте к ней быстрый доступ.

В заметку Джо добавьте два раздела:

1 **Просроченные задачи [Outstanding tasks]** Задачи, о решении которых вы договорились ранее.

2 **Новые задачи [New tasks]** Те задачи, о работе над которыми вы договариваетесь на текущей встрече.

Когда вы с Джо согласуете задачи, над которыми он берется работать, запишите их в раздел 'Новые задачи'. Сделайте четкое описание того, что ожидается от Джо для выполнения этой задачи. Завершив совещание, вставьте список действий в тело почтового сообщения для Джо и дайте этому сообщению стандартную тему (например, 'Заметки после беседы – <дата>'). Благодаря стандартному формату темы, Джо будет удобно искать в своей почте заметки с конкретного совещания.

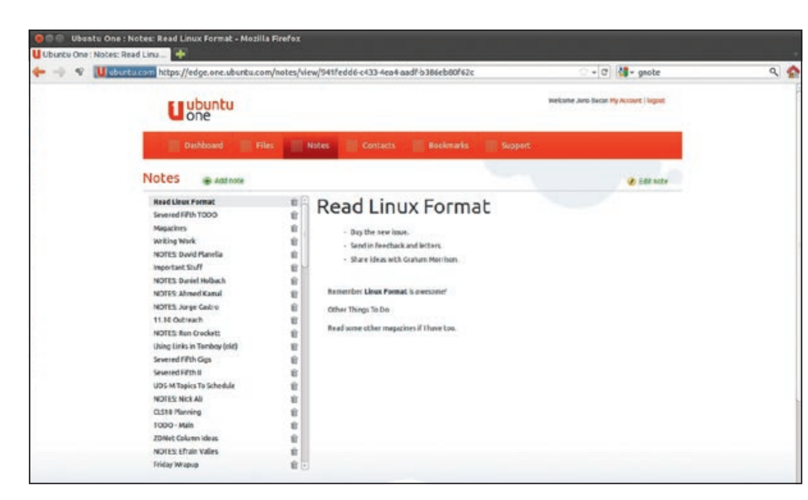
> «Вам захочется обеспечить для заметок резервные копии.»

Теперь и вам, и Джо должно быть ясно, чего от него ждут и что он должен выполнить к следующему разу. При проведении очередного совещания просмотрите все согласованные задачи. Если задача выполнена, удалите ее из списка. Если нет, перенесите ее в раздел Просроченные задачи. Следует понимать, что если задача засиделась в списке просроченных, с ней есть проблемы.

Автор одного из великолепных советов по организации – Ларри Дэвид [Larry David], создатель комедийного сериала *Curb Your Enthusiasm* [Сдержите свой энтузиазм]. Чтобы решить проблему вдохновения, посещающего нас в необычное – и часто неподходящее время, Дэвид использует черный блокнотик, где записывает все посетившие его идеи. Раз в несколько месяцев он открывает блокнот и изучает идеи, которые хотел бы развить. Прочитав об этом в интервью с Дэвидом (и принадлежа к хакерской части человечества), я захотел испытать ту же схему, применив компьютер. Я создал заметку *Tomboy* для хранения идей. В ней я не только сохраняю идеи, но и записываю свои наблюдения, интересные мысли, задумки для будущих статей и презентаций, забавные идеи, шутки, тексты песен и многое другое. Дэвид абсолютно прав: запись идей – замечательный способ не спугнуть стихийные приступы вдохновения.

Синхронизация заметок

Воплотив на практике идеи и приемы использования *Tomboy*, вы накопите внушительную коллекцию заметок, и многие из них – с ценной информацией. В идеале, вам захочется обеспечить для заметок резервные копии, чтобы не утратить их в случае отказа вашего компьютера; и неплохо было бы получать доступ к ним с любой машины. Здесь вам поможет синхронизация. В *Tomboy* встроена поддержка синхронизации заметок с разнообразными сервисами (см. врезку *Ваши заметки повсюду*). Чтобы добраться до этих опций, выберите команды Edit > Preferences и воспользуйтесь вкладкой Synchronisation [Синхронизация] для настройки синхронизации. Синхронизация через SSH сложнее, чем хотелось бы, но в справочном руководстве *Tomboy* имеются инструкции.



› Ваши заметки не прикованы к одной машине: сервис Ubuntu One позволяет читать их через web-браузер и синхронизировать с другими компьютерами.

В Ubuntu синхронизацию выполнять очень просто, пользуясь сервисом Ubuntu One. Сначала зарегистрируйтесь на http://one.ubuntu.com – бесплатная учетная запись позволяет хранить до 2 ГБ (также предлагаются синхронизация файлов, контактной информации и музыки, а также потоковая трансляция музыки). Для настройки всего этого сначала щелкните по вкладке Synchronisation и выберите Ubuntu One из выпадающего списка Service. Затем в диалоговом окне Preferences нажмите кнопку Save. В ответ на вопрос о том, хотите ли вы синхронизировать заметки сейчас, нажмите кнопку Yes. При появлении диалогового окна с сообщением об успехе синхронизации нажмите кнопку Close. В диалоговом окне Preferences есть опция задания выполнения автоматической синхронизации каждые 10 минут – уж тогда вы не потеряете ни одной заметки!

Сервис Ubuntu One не только гарантирует синхронизацию заметок на всех ваших компьютерах с Ubuntu, но и дает к заметкам доступ через web-браузер на сайте http://one.ubuntu.com.

Заключение

Tomboy дает замечательно простой и практичный метод наведения порядка в информационном хаосе нашей цифровой эры. В данной статье мы лишь слегка коснулись темы и рассмотрели только основные функции. Но в *Tomboy* множество других инструментов и функций, достойных изучения. Рекомендуем поработать с этим приложением и встроить его в рабочий процесс, чтобы с его помощью синхронизировать вашу работу и придать ей новый импульс. Наслаждайтесь! LXF



Tomboy против Gnote?

Хотя *Tomboy* пропагандируется как чертовски простое и полезное приложение, отдельные фанаты свободного ПО его отвергают, так как программа написана на C# с применением среды Mono.

Mono – противоречивое явление для сообщества: некоторых беспокоит, что Mono базируется на среде.NET от Microsoft и есть риск, что Microsoft начнет преследовать по закону разработку ПО с помощью Mono. Это беспокойство породило приложение *Gnote* (https://live.gnome.org/Gnote), клон *Tomboy*, написанный на C++ и реализующий тот же интерфейс и те же функции.

С точки зрения функциональной полноты *Gnote* чуть отстает от *Tomboy*, но ключевые возможности в нем имеются. Если Mono вас нервирует, попробуйте *Gnote*.

Сертификация в мире Linux

Для тех, кто решился избрать Linux в качестве основной профессии, **Сергей Пашала** рассказывает о доступных в России курсах обучения.



Наш эксперт

Сергей Пашала
Как сертифицированный инструктор RedHat, не первый год несет знания в массы.

Составляя резюме, многие стараются максимально приукрасить навыки, которыми они обладают. Пишут об активном участии в крупном проекте, тогда как только лишь ассистировали. Указывают на длительную работу с сетевым оборудованием, а на самом деле имеют однократный опыт. И все это ради того, чтобы работодатель заинтересовался их кандидатурой и прислал приглашение на собеседование. Возникает вопрос: а каким образом мы можем подтвердить свои знания? Как доказать работодателю, что все знания, перечисленные в нашем резюме, не являются обманом? Конечно, можно показать себя на собеседовании, но до него еще нужно добраться. Как показала практика, некоторые работодатели в первую очередь смотрят на наличие сертификации, которая иногда выступает как фильтр при отсеве неподходящих кандидатов.

В этой статье я хочу рассказать, как обстоят дела с сертификацией в мире Linux. Надеюсь, что она будет интересна не только тем, кто еще не решил, кому отдать предпочтение, но и тем, кто уже обладает рядом сертификатов. В этой статье я сделаю небольшой обзор существующих программ сертификаций и в заключение дам рекомендации по подготовке к сдаче экзаменов, с указанием ссылок на рекомендуемые программы обучения, которые помогут вам получить необходимые знания. Так как объем материала очень велик, то я решил рассмотреть только те направления, которые связанные с работой системных администраторов и инженеров. Давайте посмотрим на список вендоров, чьи сертификации будут рассмотрены:

- LPI
- RedHat
- SUSE Linux (Novell)
- Mandriva
- Ubuntu
- ALT Linux

А начну я свой обзор с широко известной на Западе некоммерческой организации, которая не только занимается сертификацией, но и активно участвует в продвижении и популяризации повсеместного использования Linux-систем. Затем мы с вами перейдем к общеизвестным вендорам, чьи дистрибутивы заслужили доверие и приобрели массу поклонников не только в корпоративной среде, но и на настольных системах. В качестве примера можно привести дистрибутив, известный своим дружелюбным интерфейсом и простой настройкой, что делает его с каждым годом все более популярным во всем мире для использования на настольных системах.

LPI: Linux Professional



LPI (Linux Professional Institute) – некоммерческая организация, основанная в октябре 1999 года, главная цель которой заключается в создании активного сообщества профессионалов и компаний, использующих Linux. Для достижения своих целей LPI работает над различными инициативами, важнейшей из которых является установление стандарта сертификации Linux-специалистов. Экзамены LPI охватывают широкий круг основных тем, связанных с Linux, но в отличие от других программ сертификации в LPI нет привязки к какому-либо «вендору». От кандидата требуется только наличие общих знаний о Linux, а не знания специфики определенного дистрибутива. С одной стороны, это усложняет подготовку, но с другой стороны, делает сертификат более универсальным и позволяет проверить компетенцию кандидата по отношению к любой Linux-системе. Как пример, можно привести вопросы, связанные с установкой и удалением программ. В них кандидата могут спросить как об опциях команд, связанных как с rpm-пакетами, так и с deb-пакетами.

На начало 2011 года программа сертификации LPI предусматривает три статуса знаний:

» LPIC-1 – Junior Level Administration [Администратор начального уровня]
» LPIC-2 – Intermediate Level Administration [Администратор среднего уровня]
» LPIC-3 – Senior Level Administration [Администратор высшего уровня]

Для получения каждого из уровней кандидату необходимо будет сдать пару экзаменов. Например, при желании получить статус LPIC-1 необходимо успешно сдать два экзамена LPI-101 и LPI-102, которые проверят следующие навыки и знания:

» Работа с учетными записями пользователей
» Настройка и управление резервным копированием и восстановление информации
» Отключение и перезагрузка системы при необходимости
» Умение устанавливать и настраивать программное обеспечение
» Умение подключать рабочие станции к сети и к Интернету.

Экзамены LPI переведены на большинство популярных языков и доступны для сдачи в тысячах различных точек по всему миру через системы тестирования Pearson VUE и Thomson Prometric.

Ссылки

» Список авторизованных партнеров LPI (Россия отсутствует): https://cs.lpi.org/caf/Xamman/partner/list_partners
» Более подробно – на официальном сайте в разделе «Обучение и сертификация». http://www.lpi.org/eng/certification

RedHat



Одним из крупных вендоров, который держит высокий уровень обучения, является RedHat. Обучение разделено на три уровня: начиная с самых базовых вещей, которые буду полезны младшим системным администраторам, и заканчивая знаниями, необходимыми для работы в качестве архитектора.

В прошлом году RedHat решил пересмотреть методику обучения, тем самым подняв уровень качества на новую ступень; но здесь есть одно «но». Теперь очень многое зависит от инструктора – от того, насколько глубоко он знает материал, и от его способности донести эти знания до слушателя. RedHat убрал большинство информации из учебников, преобразовав их в аналогию блокнота для заметок, и тем самым лишил слушателей возможности подсмотреть в учебник, если они чего-то не записали.

Как и в обучении, в программе сертификации заложено три ступени, которые определяют уровень знаний сдавшего экзамен кандидата:

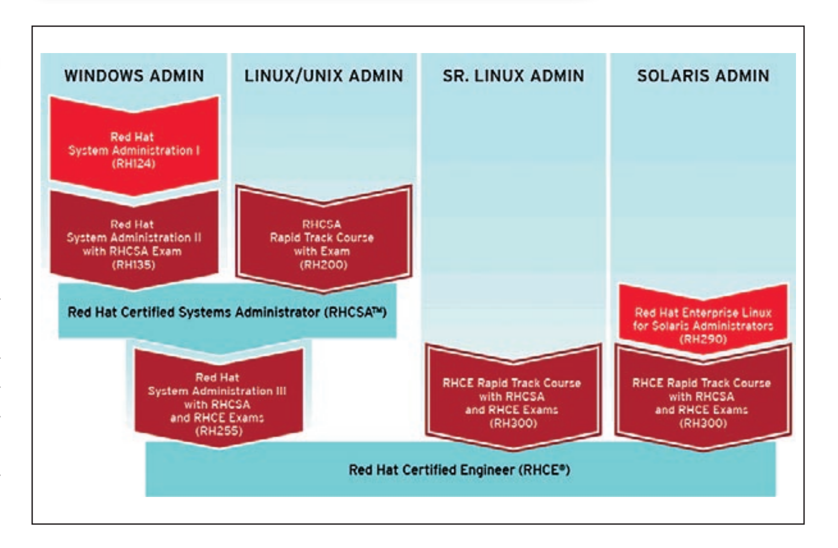
» Red Hat Certified System Administrator (RHCSA) [Сертифицированный системный администратор]
» Red Hat Certified Engineer (RHCE) [Сертифицированный инженер]
» Red Hat Certified Architect (RHCA) [Сертифицированный архитектор системы]

Во всех случаях кандидату предлагается пройти практический экзамен и показать не только теоретические знания работы с системой, но и накопленный практический опыт. В RedHat решили вернуться к уже опробованной системе и снова разбили экзамен RHCE на две части – RHCSA и RHCE, но теперь они уже не зависят друг от друга, что позволило сдавать по их по отдельности или сразу. Также был изменен срок действия сертификации: теперь он составляет 3 года со сдачи последнего экзамена. То есть, сдав сейчас RHCSA, можно пойти сдавать RHCE хоть через 3 года, и сертификат будет действителен еще в течение трех лет.

В России официальным обучением занимаются всего семь компаний: две в Санкт-Петербурге и пять в Москве

Ссылки

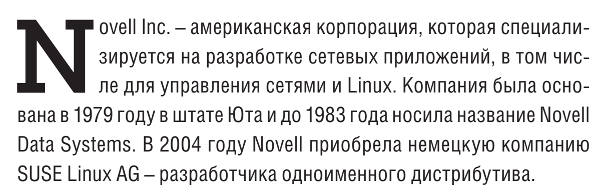
» Список авторизованных партнеров RedHat в России и странах СНГ: https://www.europe.redhat.com/training/partners.php#RU
» Более подробно – на официальном сайте в разделе «Обучение и сертификация». http://www.redhat.com/training/



› Ступени сертификации специалистов RedHat.

SUSE Linux





Novell Inc. – американская корпорация, которая специализируется на разработке сетевых приложений, в том числе для управления сетями и Linux. Компания была основана в 1979 году в штате Юта и до 1983 года носила название Novell Data Systems. В 2004 году Novell приобрела немецкую компанию SUSE Linux AG – разработчика одноименного дистрибутива.

SUSE Linux Enterprise Server (SLES) – дистрибутив операционной системы Linux, ориентированный в первую очередь на корпоративный рынок. В апреле 2011 Novell была куплена корпорация Attachmate, и торговая марка SUSE Linux перешла к ней. Выпускается для серверов стандартной архитектуры, машин IBM pSeries (POWER) и мейнфреймов zSeries, с интервалом в 24–36 месяцев.

На начало второго квартала 2011 года доступны три ступени сертификации:

- Certified Linux Administrator 11 [Сертифицированный администратор Linux]
- Certified Linux Professional 11 [Сертифицированный профессионал Linux]
- Certified Linux Engineer 11 [Сертифицированный инженер Linux]

Первая ступень сертификации практически полностью соответствует LPIC-1, и для ее достижения достаточно сдать LPI-101 и LPI-102 либо всего один экзамен от Novell, аналогичный по содержанию. Две последующие ступени требуют знания специфических особенностей дистрибутива SUSE Linux, и для достижения каждой из них необходимо пройти практический экзамен.

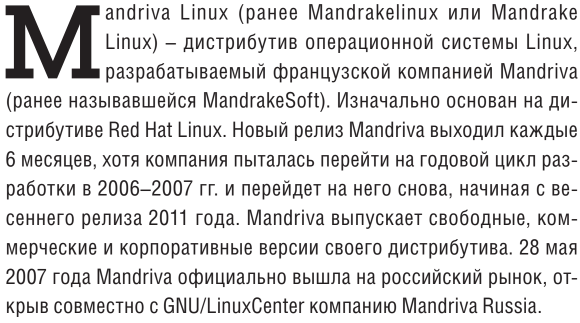
Обучение доступно в двух городах: Санкт-Петербурге и Москве.

Ссылки

- Список авторизованных партнеров Novell в России и странах СНГ: http://www.novell.com/partnerlocator/locator.do?country=RU
- Более подробно – на официальном сайте в разделе «Обучение и сертификация». http://www.novell.com/training/certinfo/

Mandriva





Mandriva Linux (ранее Mandrakelinux или Mandrake Linux) – дистрибутив операционной системы Linux, разрабатываемый французской компанией Mandriva (ранее называвшейся MandrakeSoft). Изначально основан на дистрибутиве Red Hat Linux. Новый релиз Mandriva выходил каждые 6 месяцев, хотя компания пыталась перейти на годовой цикл разработки в 2006–2007 гг. и перейдет на него снова, начиная с весеннего релиза 2011 года. Mandriva выпускает свободные, коммерческие и корпоративные версии своего дистрибутива. 28 мая 2007 года Mandriva официально вышла на российский рынок, открыв совместно с GNU/LinuxCenter компанию Mandriva Russia.

Согласно данным сайта www.mandriva.ru, обучение может проводится как по программе LPI, так и по собственной программе, которая нацелена на сдачу следующих экзаменов:

- MCU: Mandriva Certification For Advanced Users [Сертификат продвинутого пользователя]
- MCA: Mandriva Certification For Networking Administrators [Сертификат сетевого администратора]

Справедливости ради стоит сказать, что и на западном сайте есть только упоминание об онлайн-тренинге, а никакой информации об экзаменах и сертификации нет. В России программа обучения и сертификации базируется на курсах и экзаменах LPI, причем доступны онлайн-курсы.

Ссылки

- Полный список авторизованных партнеров найти не удалось.
- Более подробно – на официальном сайте в разделе «Обучение и сертификация». http://etraining.mandriva.ru/

Ubuntu



Ubuntu – один из самых популярных дистрибутивов Linux на сегодняшний день; его разработкой занимается компания Cannonical. Изначально целью данного дистрибутива были удобство и простота использования, и с каждым новым релизом разработчики все ближе придвигаются к своей цели. Это и много другое, что делает компания Canonical, способствует еще большей популярности данного дистрибутива среди пользователей настольных систем. За основу Ubuntu разработчики взяли не менее популярный проект под названием Debian.

Для желающих пройти обучение компания Cannonical предлагает курсы двух типов: классический тренинг в классе и online-обучение, которое будет интересно всем, кто не может посещать занятия или желает сэкономить. В России на данный момент официально проводит обучение только компания R-Style.

В 2006 году компания Cannonical совместно с LPI анонсировали сертификацию UCP (Ubuntu Certified professional [Сертифицированный профессионал Ubuntu]), который должен подтвердить базовые знания системных администраторов. Чтобы получить сертификацию UCP, кандидату нужно успешно сдать экзамены LPI 101, LPI 102 и Ubuntu 199, доступные в центрах тестирования VUE и Prometric по всему миру.

По непонятным автору причинам информация о сертификации с официального сайта www.ubunutu.com была убрана. Все ссылки из внешних источников, ранее указывавшие на раздел сертификации, теперь ведут в раздел обучения.

Ссылки

- Список авторизованных партнеров Ubuntu в России и странах СНГ: http://webapps.ubuntu.com/partners/training/
- Более подробно – на официальном сайте в разделе «Обучение и сертификация». http://www.ubuntu.com/support/training

ALT Linux



Дистрибутивы ALT Linux (Альт Линукс) – семейство, являющееся отдельной ветвью развития русского Linux, выпускаемых компанией «Альт Линукс» и ее партнерами на основе разработок русскоговорящей команды разработчиков ALT Linux Team. Большинство дистрибутивов Альт Линукс доступны для свободного скачивания.

В настоящее время компания ALT Linux выдает сертификаты двух видов:

» Сертифицированный специалист ALT Linux

» ALT Linux Workstation administrator [Администратор рабочей станции ALT Linux]

Для получения сертификата кандидату необходимо самому собрать стенд и выполнить на нем задачи, которые ему вышлют по электронной почте. Как только задачи будут выполнены, кандидату необходимо будет предоставить специалисту ALT Linux доступ на стендовый сервер по **ssh** для проведения экспертизы. На сайте разработчика есть упоминание, что экспертиза занимает 5 рабочих дней. Также, в случае успеха, кандидату будет назначено собеседование, которое проводится через **jabber** и продолжается не более 2-х часов.

Обучение доступно в шести городах: Санкт-Петербурге, Москве, Самаре, Калининграде, Ельце и Благовещенске.

Ссылки

» Авторизованные партнеры ALT Linux в России и странах СНГ: http://www.altlinux.ru/partners/partners-list/trainers/

» Более подробно – на официальном сайте в разделе «Обучение и сертификация». http://www.altlinux.ru/training/courses/

Советы инструктора

Как вы уже успели заметить, только RedHat, Novell и ALT Linux занимаются разработкой и продвижением своих программ обучения; все же остальные либо базируются на LPI-курсах, либо вообще не имеют таких программ. Курсы, разрабатываемые в LPI, дают общий набор знаний, который позволяет слушателю в дальнейшем чувствовать себя комфортно в любом из известных дистрибутивов. Тренинг же RedHat и Novell несет свой набор знаний, который ничуть не хуже обучения по программе LPI, но в дополнение к этому несет знания, которые помогут вам научиться использовать так называемые вендоро-зависимые решения. И если вам по роду деятельности необходимо работать именно с системами этих фирм, посещение курсов в большинстве случаев значительно облегчит жизнь.

> «Как бы ни казалось, что на курсах ничего нового, это неверно.»

С сертификацией почти такая же картина, как и с обучением. В большинстве случаев каждый из вендоров придерживается многоступенчатой системы сертификации, которая охватывает различные аспекты работы с Linux. Но все они в основном строятся на сертификации LPI, в которой проверяется общий набор знаний. В отличие от LPI, RedHat пошел своей дорогой и проводит практический экзамен. В результате статус RHCE до сих пор воспринимается как эталон качества и подтверждает, что владелец данного сертификата владеет не только теоретическими знаниями, но и имеет большой багаж практического опыта, без которого возможность получить данный статус сводится к нулю.

Всем, кто желает получить ту или иную сертификацию, я бы посоветовал в первую очередь обратить внимание на курсы, нацеленные именно на подготовку к сдаче экзаменов. И как бы ни казалось, что курсы очень дороги и в них для вас нет ничего нового, на самом деле это неверно. На курсах вы не только освежите свои знания, но и узнаете о решениях, которые применяет тот или иной вендор, и впоследствии это избавит от неприятных минут на экзамене. Если же вы только начинаете изучать Linux, то курсы позволят за кратчайший срок получить больший объем знаний, чем при чтении книг и изучении документации, что впоследствии позволит намного легче продвигаться в изучении данного направления. LXF

Linux: Вопросы праистории

В год юбилея уместно еще раз вспомнить историю Linux и людей, к ней причастных, считает **Алексей Федорчук**.

Данный цикл заметок сочиняется в преддверии знаменательной даты — двадцатилетия с того дня, когда программа эмуляции терминала, разработанная двадцатилетним студентом, неожиданно превратилась в ядро новой операционной системы, получившей имя Linux.

Правда, какой это был день, мнения расходятся. Одни историки вопроса считают датой рождения Linux'а 25 августа 1991 года, когда Линус Торвальдс отправил в список рассылки разработчиков ОС MINIX свое ставшее знаменитым письмо о том, что он разрабатывает новую бесплатную ОС, которая скоро будет готова. Другие стоят за 17 сентября 1991 года – день, когда Линус разместил исходный текст своей разработки на общедоступном ftp-сервере, о чем оповестил заинтересованных лиц, в первую очередь пользователей MINIX. И предложил присоединяться к проекту.

Мы с Кириллом Степановым посоветовались и пришли к единогласному, хотя и волевому, решению: вторая точка зрения более обоснованна. Во-первых, именно в этот день Linux получил свое имя – по имени каталога, где лежали исходники ОС на ftp-сервере. А во-вторых, с этой даты начинается внедрение в софтверную индустрию того метода, который позднее Эрик Реймонд назовет «базарным», но который по справедливости следовало бы наречь именем его изобретателя – Тома Сойера: Линус лишь подхватил, расширил, укрепил и закалил в боях его идею.

В-третьих же, мы надеемся, что к 17 сентября 2011 года номер **LXF** со статьей, посвященной рождению Linux'а, его предпосылкам и последствиям, выйдет из типографии. И даже имеет шанс попасть в руки некоторых читателей. Оставшееся же до славной даты время мы посвятим вопросам праистории и предыстории любимой ОС.

› **В 1982 году журнал Electronics International присвоил Ричи и Томпсону награду за высшие технологические достижения.**

Истоки и составные части

Было бы заманчиво свести происхождение Linux'а к трем истокам и трем составным частям, по завету дедушки Ленина. Однако реальная история чего бы то ни было гораздо сложнее (и интересней), чем нам внушали классики марксизма и их переложители: с какой стороны ни смотри, а истоков и составных частей Linux'а оказывается куда больше трех. Да и не всегда однозначно скажешь, где кончаются первые и начинаются вторые. Можно, впрочем, внять мнению резонных одесситов и признать, что составные части начинаются именно там, где кончаются истоки – рассмотрим их одним списком, а уж читатель сам решит, где одно перетекает в другое. Единственная вольность – я немного упорядочил этот список согласно своим представлениям о хронологии событий.

Итак, приступаем к подсчету:

1. Академическая и университетская Computer Science эпохи «больших машин» – в частности, работы по искусственному интеллекту.
2. Сеть ARPANET и все сопряженные с ней явления, из которых вырос в итоге современный массовый Интернет. Ибо только нынешние «одноглазнеги» полагают, что Интернет затевался для того, чтобы они могли сидеть «ффконтакте».
3. Корпоративный Unix и организации по упорядочиванию его стилей, т. е. стандартизации.
4. Берклиада – история, полная драматизма, в которой, как и в поэме Гомера, впервые переплелись перечисленные выше пряди. Но о ней будет подробно говориться в следующей заметке.
5. Общественные движения – Open Source Software и Free Software. Здесь я немного отступаю от хронологии: второе формально предшествовало первому. Но de facto первое явление существует как минимум со времен Ньютона и Лейбница.
6. Эволюция аппаратных платформ – от первых интерактивных рабочих станций до «народных» x86-совместимых компьютеров.

Как видите, получается вдвое больше сакральной троицы, которую так любят Бог и большевики. И я даже не уверен, вспомнил ли я все истоки и все составные части… ну, не страшно: нам удастся осветить далеко не все и из перечисленного.

О некоторых же истоках я умолчал сознательно, ибо нет у меня права о них говорить – например, об эпохе Eniac'а и БЭСМ'ов. Надеюсь, что

Кто-нибудь услышит,
Снимет и напишет,
Кто-нибудь помянет...

тех ребят. Они заслуживают этого не меньше, чем создатели первого реактора... Да и чем многие другие, которых тоже еще не помянули.

А пока займемся нашими истоками и составными частями.

Computer Science и ARPANET

История академической Computer Science уходит в начало 60-х, время появления первых компьютеров, способных к интерактивной работе. Хотя они далеко еще не были персоналками – но ведь ранее машины, как помнят читатели «Понедельника», почему-то начинавшегося тогда «в субботу», работали исключительно в режиме пакетных заданий. И повлиять на это было не по силам не только магам вроде Кристобаля Хозевича и Федора Симеоновича, но даже всесильным научным администраторам, товарищам Модесту Матвеевичу и Лавру Федотычу.

Место же зарождения этой науки условно определим как крупнейшие американские университеты – Массачусетский технологический институт (MIT), Йель, Университет Карнеги-Меллона, Стэнфорд, Калифорнийский университет Беркли. Историческим центром этого движения долгое время была лаборатория искусственного интеллекта MIT (MIT AI – Artificial Intelligence). В недрах MIT AI родился, судя по многим свидетельствам, и термин «хакер» – так называли друг друга те, кто способен был «врубиться» в компьютерные науки. Но эта тема столь жевана и пережевана, что на ней мы останавливаться не будем.

Работы же по созданию отказоустойчивой правительственной связи США, как нетрудно догадаться, начались по инициативе Министерства обороны этой страны. Ибо имели целью создание надежной системы передачи информации на случай советского ядерного удара. Финансирование осуществлялась через ARPA – Агентство передовых исследовательских проектов (Advanced Research Projects Agency), которое позднее, без лишнего лицемерия, было переименовано в DARPA, с добавлением слова Defense (в данном контексте – Оборонных проектов). Запомним последнюю аббревиатуру – позднее эта организация сыграет немалую роль в нашей предыстории.

Непосредственная реализация системы связи была возложена на ряд американских университетов – Калифорнийский, Университет штата Юта, Стэнфорд: оказалось, что кроме университетских хакеров из сферы Computer Science разрабатывать и поддерживать ее было попросту некому. А эти «ребята, за ту же зарплату», не только выковали электронный щит своей Родины в виде сети ARPANET (по имени организации-кормильца), но, будучи истинными учеными, воспользовались случаем в интересах науки. А именно – наладили

бесперебойные каналы обмена информацией между своими Alma mater, создав таким образом сообщество ARPANET – прообраз грядущего Интернет-сообщества.

Первый сеанс связи в рамках проекта ARPANET состоялся 29 октября 1969 года в 21 час по местному времени и оказался не вполне удачным: передались только три символа, после чего сеть рухнула. Однако уже через два часа работоспособность ее была восстановлена (учитесь, нынешние провайдеры), и передача завершилась успешно. Интересно, что контроль передачи осуществлялся почти тем же методом, который изображен в фильме «Волга-Волга» – не с помощью рупора, конечно (от Лос-Анджелеса до Пало-Альто докричаться проблематично), но по телефону.

Сеть ARPANET очень быстро охватила не только университеты, вовлеченные в ее разработку, но и многие другие учебно-научные заведения Америки, а потом и сопредельных стран, став международной коммуникационной магистралью для обмена научной информацией. Правда, скоро эту роль перехватила сеть Национального научного фонда США (NSF – National Science Foundation) – NSFNet, с большей пропускной способностью. Именно на ее базе и был создан современный Интернет.

Рождение Unix

Зарождение Unix также связано с появлением компьютеров, способных на интерактивный режим, что создало предпосылки к разработке систем разделения времени, допускающих как бы одновременное исполнение нескольких задач [time sharing] – они пришли на смену машинам, работавшим исключительно в пакетном режиме. Одной из первых таких систем была CTSS [Compatible Time Sharing System].

Без академической составляющей, представленной в данном случае MIT, не обошлось и здесь. В развитие CTSS в 1965 году фирмами AT&T и General Electric вместе с MIT был начат проект по созданию истинно многозадачной и многопользовательской системы, получившей имя Multics. По замыслу она была столь прогрессивна, что в те времена оказалась неосуществимой, и в 1969 году проект закрыли, оставив его участникам тоску по интерактивной работе и идею разделения времени, вскоре вылившуюся в Unix.

Хотя сама ОС Unix вышла из корпоративных недр компании AT&T, сотрудниками которой являлись бывшие участники проекта Multics, это ни в коей мере не была корпоративная разработка: Кен Томпсон [Ken Thompson] и Деннис Ричи [Dennis M. Ritchie] разрабатывали ее для собственных нужд. Это был первый в истории IT пример создания «системы для себя», в противоположность, например, системе VAX/VMS от фирмы DEC, претендовавшей на звание «системы для всех».

Правда, понятие «все» в случае с VAX/VMS охватывало весьма узкий круг – даже не столько лиц, сколько организаций. Тот факт, что система VAX/VMS разрабатывалась не для личного пользования, наложил отпечаток не только на нее: это предопределило судьбу ее прямого потомка – Windows NT/etc. Unix же разрабатывался на голом энтузиазме, без особой поддержки дорогих руководителей, причем на аппаратном обеспечении, подлежащем списанию в утиль.

Впрочем, усилия разработчиков были оценены должным образом, и достаточно быстро: в 1983 году Томпсону и Ричи присудили премию Тьюринга – это самая престижная награда в информационной сфере, по значимости сравнимая с премией, учрежденной некогда Альфредом Нобелем «за выдающиеся научные исследования, революционные изобретения или крупный вклад в культуру или развитие общества». Что поделать – Нобель не мог и предполагать, что информационные технологии окажут на развитие общества не меньшее влияние, чем изобретенный им динамит. А в 1999 году Томпсон и Ричи удостоились одной из высших наград государства, гражданами которого они являются: Национальной медали в области технологий (National Medal of Technology, ныне – National Medal of Technology and Innovation).

На дальнейшую судьбу Unix огромное влияние оказали юридические коллизии тогдашнего текущего момента. Незадолго до создания этой системы корпорация AT&T подверглась антимонопольному преследованию, в результате чего претерпела поражение в правах – на деятельность ее был наложен ряд ограничений. В частности, она не имела права торговать программными продуктами, в число коих попадала и новорожденная Unix.

Разумеется, материнская корпорация постаралась пристроить к делу детище своих сотрудников – так, Unix с его инструментарием использовался в AT&T для подготовки технической и патентной документации. А это, кстати, типичная пользовательская задача; вот и говорите теперь, что Unix не пригоден для применения конечными пользователями.

> «Unix зажил самостоятельной жизнью в университетской среде.»

Однако, как уже было сказано, в силу юридических ограничений AT&T не могла сделать из Unix коммерческий продукт. И потому исходники этой системы, начиная с 1974 года, стали распространяться в университетах – в образовательных, как это тогда задумчиво называлось, целях. На условиях по тем временам достаточно либеральных, в том числе и просто явочным порядком, лично Брайаном Керниганом [Brian Kernighan] – люди с психологией сталинских наркомов, которые имели мужество сказать «под мою ответственность», встречались не только в Советском Союзе...

Передача Unix в университетские структуры не была свободным распространением в том смысле, который вкладывается ныне в понятие FOSS. Хотя система – точнее, тогда еще не более чем ее прототип – и передавалась в исходных текстах с правом их изучения, модификации, доработки и прочего потрошения.



› **Первый в мире UNIX-хост DEC PDP-7. Обратите внимание — на переднем плане слева первый в мире /dev/tty0.**

Однако, во-первых, все эти действия требовали обладания лицензией на исходный код Unix, которая передавалась AT&T вместе с ней самой и ее исходниками, но – за деньги, хотя и не очень большие по масштабам американских организаций середины 70-х годов прошлого века. В личное же пользование лицензии на Unix тогда еще не приобретали.

А во-вторых, и это главное, условия лицензии не допускали дальнейшего свободного распространения ни системы целиком, ни отдельных ее компонентов, содержащих исходный код Unix. Что, собственно, и создало сюжет дальнейшей детективно-технологической интриги.

Однако до развертывания грядущего технологического детектива было еще далеко. А пока университеты радостно приобщались к новой операционной системе, в которой были реализованы все передовые идеи того времени, к тому же в принципе способной функционировать практически на всем спектре тогдашнего оборудования. Напомню, что речь идет о середине 70-х годов прошлого века: Стив Джобс еще не помышлял о продаже калькулятора и использовал родительский гараж по прямому назначению, а Билл Гейтс не освободил мир своим MS DOS'ом от засилья CP/M.

Выйдя за стены Bell Labs, Unix зажил самостоятельной жизнью, прочно окопавшись в той же университетско-академической среде Computer Science. Одним из ее центров в данном случае оказался Калифорнийский университет Беркли – учреждение, известное всем, интересовавшимся историей как точных наук, так и их влиянием на нашу жизнь.

Получив, благодаря профессору Бобу Фабри [Bob Fabry], в 1974 году ОС Unix вместе с ее исходниками и лицензией на их использование, университет Беркли поддержал и развил традицию «систем для себя», свойственную первозданному Unix. Но об этом – в следующей заметке. LXF

Что за штука... Calligra?

Думаете, вы знаете *KOffice*? Э, нет. «Другой» офисный Linux-комплект откинул букву «К» и ищет славу кроссплатформенного, как засек **Энди Ченнел**.

» *Calligra?* Что-то знакомое. Смахивает на искуственные слова — вроде они что-то значат, а что — непонятно...
Так и есть. Название происходит от слова «каллиграфия» (хотя сама каллиграфия здесь ни при чем) и демонстративно избегает заглавной и вообще любой буквы К. Фактически это новое название офисного комплекта *KOffice*.

» Так это просто смена вывески — отказ от буквы К и освежение экранов загрузки приложений?
Первоначально процесс выглядел именно так, но по сути это ответвление. То есть группа разработчиков, недовольных состоянием проекта, решила взять существующую кодовую базу и пойти тем путем, которое считала верным. Позднее были попытки устранить разветвление (как в *Compiz*), было углубление различий (подобно *OpenOffice.org* и *LibreOffice*), а в итоге получилось два конкурирующих проекта.

> «Концепция комплекта вышла за пределы офисного ПО.»

» Что потеряли *KOffice/Calligra*?
К сожалению, потери были с обеих сторон. Группу участников проекта не устраивало направление развития, и они решили отделиться – довольно необычно, поскольку они составляли большинство. По разным личным и политическим причинам решено было отказаться от упражнений в именовании с буквой «К» и принять концепцию разветвления, чтобы переоценить состояние ПО и цели его развития в рамках сообщества разработчиков и пользователей. Хотя это и не было глобальной катастрофой в мире ПО, некоторые участники утратили ориентиры и растерялись. Отколовшаяся группа продолжает выпускать *KOffice*, одновременно работая над новым проектом *Calligra* – вот так сложно переплелись ветвление и смена названия. Лучше не ломать голову, а просто считать этот процесс стадией эволюции.

» Изменились ли названия отдельных компонентов?
По большей части. Некоторые, например *Krita*, остались неизменными, другие переименованы. Ведущий разработчик *Krita* Боудевейн Ремпт [**Boud**ewijn Rempt] сказал, что название *KOffice* давно стало обузой, и перемены назрели по многим причинам: «[Название] предполагает жесткую зависимость от рабочего стола KDE (а не от платформы), тогда как *KOffice* и его компоненты работают повсюду – на Gnome, Windows, Maemo, MeeGo и т. п.». Кроме того, концепция комплекта вышла за пределы традиционного офисного ПО, и пользователям все труднее объяснить присутствие в комплекте креативных приложений типа *Krita*. Существует и проблема восприятия. «За *KOffice* закрепилась недобрая слава, – продолжал Ремпт, – но за последние два года, несмотря на множество препятствий, проделана огромная работа по повышению надежности и унификации (например, созданы фильтры импорта), и негативная репутация вредит проекту».

» Кажется, дело не только в смене названия.
Да. Ремпт характеризует новое движение как шанс переломить ход событий, оживить сообщество и создать проект, где «людей будут уважать за их вклад в общее дело, а название станет символом креативности, элегантности и надежности». По его словам, предполагается выпустить серию «стабильных» версий для исправления ущерба, нанесенного комплекту в последние годы, и ко времени выпуска первой официальной версии проекту *Calligra* будет чем удивить потенциальных пользователей.

» Что это означает для рядового пользователя?
Лидеры проекта выделяют две целевые группы пользователей: разработчиков и пользователей, работающих на нескольких платформах. Для разработчиков есть идея создать нечто вроде *WebKit* с прицелом на создание, редактирование и отображение широкого диапазона форматов. С помощью этой платформы можно будет создать любое приложение, от апплета для отображения файлов *OO.o Impress* на телефоне Android до полноценного офисного комплекта для OS X.

Чтобы продемонстрировать достижения проекта, создаются базовые комплекты приложений; два из них почти готовы. Один – стандартный офисный комплект, ранее известный как *KOffice*, состоит из *Words*, *Tables* и *Stage* (бывшие *KWord*, *KSpread* и *KPresenter*) плюс *Kexi*, *Karbon14* и *Plan* для остальных целей. Второй комплект – *Calligra Mobile* для Maemo и его отпрыска MeeGo: он состоит из *Words*, *Tables* и *Stage*.

Ремпт говорит, что основные усилия разработчиков сейчас направлены на «настольную» версию; однако проект уже перерос в «платформу», и успехи одного направления положительно сказываются на всех остальных. «Любое улучшение движка улучшает все приложения, построенные на нем – рабочие столы, мобильные устройства, планшеты. Я своими глазами видел версию *Calligra Tables*, работающую внутри web-браузера благодаря Lighthouse на *Qt*», рассказывает он.

» А кто работает над проектом — участники сообщества?
Сейчас проект собрал вокруг себя устойчивое сообщество разработчиков. В «период полураспада» *KOffice* многие талантливые программисты разочаровались и покинули лагерь, однако идея платформы оказалась жизнеспособной, в дело вступили другие люди и компании, и проект в целом преуспел. Некоторые из тех, кто в свое время ушел, вернулись вдохновленные и полные свежих идей. Пришли и новые участники. Группа студентов из Бангалора создала модуль для системы, позволяющей мобильным пользователям переводить документы с помощью Google Translate; Nokia добавила поддержку открытого формата документов (ODF), принятого в *OO.o* – очень ценный вклад.

» А разве Nokia еще не вошла в состав Microsoft? Будет ли это работать?
Похоже, внезапная любовь Nokia к продуктам Microsoft скажется на ее участии в проекте – однако компания известна тем, что поспевает всюду. По словам Ремпта, компании, подобные Nokia,



обычно берут на себя малозаметную часть разработки приложений – например, создание фильтров импорта и экспорта для доступа пользователей к документам *Microsoft Office*: «Nokia внесла масштабный вклад, вполне заслуживающий похвалы. И в данном проекте Nokia ее получает. В целом степень вовлеченности компании поразительна – доработка фильтров импорта старых двоичных форматов *Microsoft Office*; полная разработка фильтров импорта OOXML; многочисленные улучшения обработки ODF; масса исправленных ошибок в центральном движке и движке форматирования текста, а еще крупные участки *Calligra Mobile*».

Рempт добавляет, что по мере свертывания активности Nokia ее задачи берут на себя новые участники проекта, как на этапе зарождения *WebKit*. «Наш стенд App-Up (на Mobile World Congress в Барселоне) демонстрировал прототип *Calligra* для планшетов и вызвал большой интерес – некоторые компании включили *Calligra* в свои приложения для вертикальных рынков, например, приложения EPD для портативных устройств, и один из крупных производителей мобильных телефонов работает сейчас над мобильным офисом на основе *Calligra*. Но не всем пока еще понятно, что для успеха *Calligra* нужно сплотиться и вложить максимум усилий в создание основы».

» Будущее обещает быть ярким. Когда можно будет потрогать новое ПО?

Ремпт нисколько не сомневается в успехе проекта, поскольку он построен на принципах свободного ПО; тем не менее проекту требуется помощь – в любой форме, от фактического участия в программировании и устранении ошибок до тестирования и составления документации. Текущая версия (пока что придерживающаяся старого названия, *KOffice*) доступна в обычных источниках и свидетельствует об эволюции проекта, хотя на данном этапе ему недостает той завершенности, о которой мечтают программисты и разработчики. LXF



По рецептам доктора Брауна

Д-р Крис Браун
Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Эзотерическое системное администрирование из причудливых заворотов кишок серверной
















Восприятие как данности

Молодые люди вокруг меня сегодня принимают как данность то, что с мобильного телефона можно открыть Facebook, Twitter и YouTube, и большинство из них, вероятно, даже не задумывается о технологиях в основе этого. Должен ли я обвинять их в высокомерии или невежестве, или это вполне обычное поведение?

Мое собственное отношение к персональным компьютерам, Интернету и мобильникам – иное, потому что они возникли при мне. Я помню времена, когда их не было, и видел, как они развивались. В конце концов, первый пакет прошел по сети с пакетной коммутацией всего 40 лет назад.

Я более или менее помню те времена, когда не было и телевидения. Конечно, будучи мальчонкой, я лишь смутно понимал, что эта технология – на заре развития, и, как у многих моих ровесников, первой виденной мною телепередачей была коронация Елизаветы II в июне 1953 года: я смотрел ее в доме соседа.

Зато радио я принимал как данность; и хотя дом моих родителей ломился от книг, я и не задумывался о временах, когда не было технологий, позволяющих печатать их дешево и нести написанное в массы.

Отрыв поколений

Вернемся к юнцам с пальцами, прикипевшими к клавиатурам смартфонов: не стоит думать о них плохо. В каждом поколении сегодняшние люди воспринимают вчерашние технологии как данность. Но им и в голову не приходит, что завтрашние люди будут таким же образом воспринимать сегодняшние технологии.

Если после всего сказанного вам захотелось узнать побольше об истории Интернета, прочтите книгу Нейла Ренделла [**Neil Randall**] *«Душа Интернета: боги и обитатели сети и связь со всем миром»* [**The Soul of the Internet**: Net Gods, Netizens and the Wiring of the World]. Она была опубликована в 1997 году, но все еще доступна на Amazon.

Большой спрос

Тренды трудоустройства На рынке труда повышается спрос на экспертов в Linux. По какому дистрибутиву?

Спрос на программистов COBOL снижается. Вы спросите, откуда я знаю? А мне попался сайт по поиску работы (www.indeed.com), позволяющий строить графики спроса на рабочую силу за последние шесть лет на основе ключевых слов в описаниях вакансий.

Я потратил счастливые полчаса на построение графиков всего, что ни измыслил. Спрос на промышленных альпинистов невысок, но это не беда: высоты я боюсь. А вот спрос на специалистов по Linux неуклонно растет. Один из множества моих графиков отражает относительную популярность Red Hat, Fedora, SUSE и Ubuntu на рынке труда (см. на рис.). Этот график в общем отражает предпочтения в дистрибутивах тех людей, которые посещают мои учебные занятия: у них доминирует Red Hat. Я попытался построить график по аббревиатурам сертификации RHCE (Red Hat), NCLP (Novell) и UCP (Ubuntu), и Red Hat снова опережает всех, а UCP практически на нуле.

› Графики трендов с сайта Indeed отражают превосходство Red Hat.

Несмотря на обнадеживающие участки роста на графиках, значения на вертикальных осях невелики (даже Red Hat достигает максимума на 0,15 %), и чтобы было с чем сравнивать, я добавил еще несколько ключевых слов. На все графики здесь нет места, но ниже я привел несколько текущих показателей. Зайдите на сайт и взгляните сами. Кто знает, вдруг вы найдете себе работу!

Ложь, гнусная ложь и статистика

Цифры www.indeed.com (по-видимому) получены механическим поиском по ключевым словам в описаниях вакансий, поэтому вхождение «Linux», например, не означает, что знание Linux – основное требование данной вакансии. Сайт http://jobindex.recruit.net при построении графиков позволяет задавать ключевые слова, страны и даже отдельно взятые города.

Поисковое слово	Доля запросов
Microsoft	10,0 %
Windows	4,7 %
Java	3,5 %
SQL Server	2,8 %
Linux	2,4 %
Perl	1,2 %
MySQL	0,8 %
Python	0,6 %
Ruby	0,4 %

Кроссплатформенные приложения

Mono Посмотрим, как Mono, среда времени выполнения и разработки .NET, конкурирует с предложениями от Microsoft.

Постоянные читатели этой рубрики (а мое самолюбие уверяет меня, что они есть) знают, что я зарабатываю на хлеб преподаванием Linux, но у меня есть и мрачная тайна – я преподаю еще и C# и .NET! Меня часто спрашивают, почему я нахожусь и в лагере открытого ПО, и в лагере Microsoft, и, честно говоря, откажись я от одного или от другого, это бы освободило немного столь необходимого места в моей голове. Но и то, и другое приносит работу, и отказаться от нее трудно.

По сути, .NET – среда времени выполнения приложений с огромной коллекцией библиотек классов. Вместе эти возможности позволяют довольно сильно изолировать приложения друг от друга и от ОС. Обычно приложение .NET не выполняет прямых вызовов функций ОС, позволяя – как минимум, в теории – скомпилировать и запустить приложение в любой ОС с поддержкой среды времени выполнения .NET.

Основной язык разработки в .NET – C#. Это объектно-ориентированный язык, во многом схожий с Java и синтаксически тесно связанный с семейством языков C. Популярен также Visual Basic, и существуют компиляторы .NET для множества других языков, включая Perl, PHP, Python, Fortran, COBOL и другие. Эти компиляторы формируют код на промежуточном языке (точно так же, как компиляторы Java формируют байт-код Java), а в среде .NET есть компилятор, который «на лету» преобразует промежуточный язык в машинный код. Эта архитектура позволяет довольно легко интегрировать различные языки. Можно писать разные классы на разных языках (в одном и том же приложении) и даже написать базовый класс (например) на C# и унаследоваться от него в производном классе, написанном на VB.

> «Растет группа людей, использующих средства разработки Linux.»

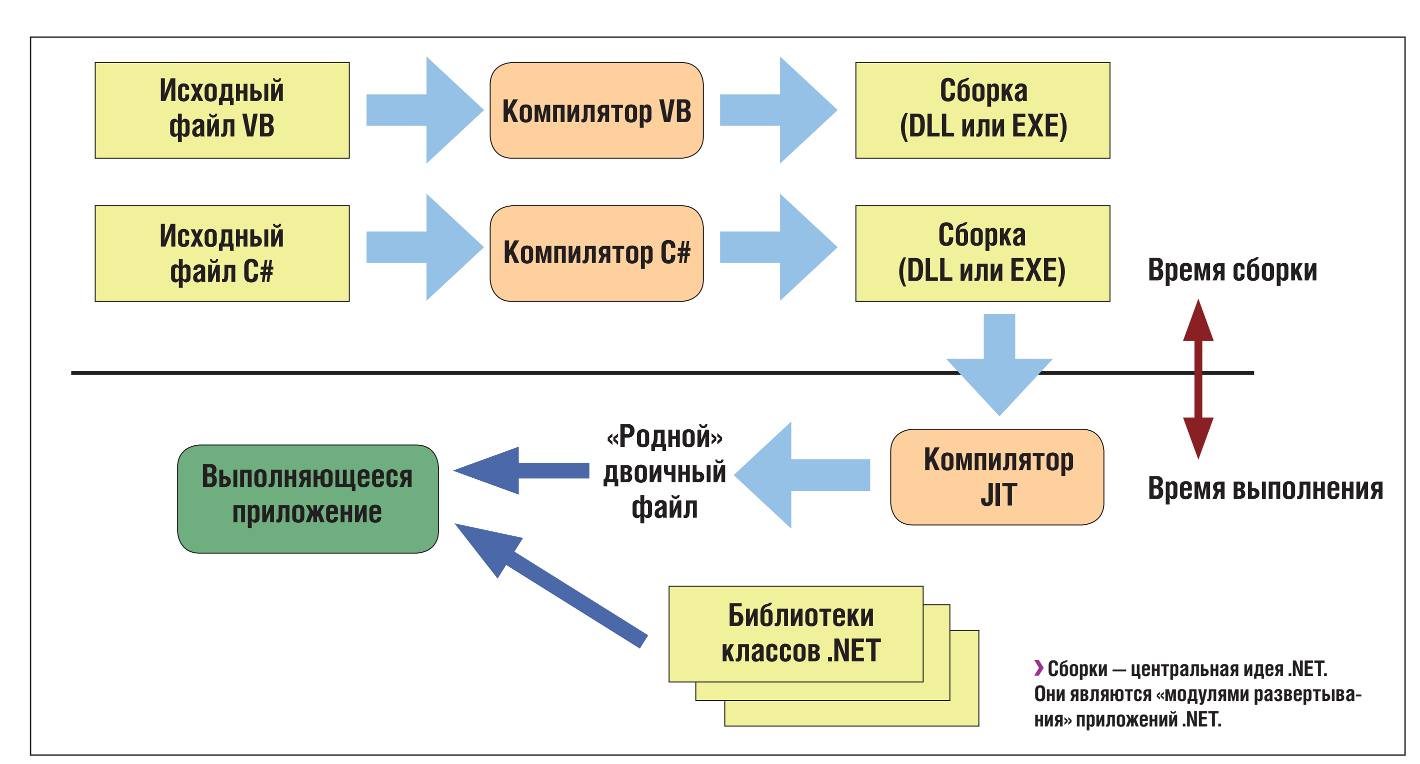
Архитектура .NET

Приложение .NET вступает в жизнь как исходный файл на C#, VB или любом другом языке, имеющем компилятор .NET. Компилятор генерирует сборки (файлы EXE или DLL) в виде кода на языке Common Intermediate Language (промежуточный язык обмена, по типу байт-кода Java), а тот на стадии исполнения преобразуется в машинный код компилятором «на лету» [JIT – just-in-time] в среде времени выполнения .NET. Загрузчик сборки также загружает заранее определенные сборки, нужные приложению, из библиотеки классов .NET.

Подавляющее большинство разработчиков пишет приложения в .NET, пользуясь средствами Windows как для разработки (*Microsoft Visual Studio*), так и для развертывания. Однако растет группа людей, использующих средства разработки Linux (*MonoDevelop*) и среду времени выполнения .NET в Linux (Mono). Приложения Linux, использующие Mono, о которых вы могли слышать, включают *Tomboy* (менеджер заметок), *Beagle* (средство индексирования и поиска), *F-Spot* (программа для управления фотографиями) и *Banshee* (медиа-проигрыватель). Гораздо более полный список см. на сайте www.mono-project.com/Software.

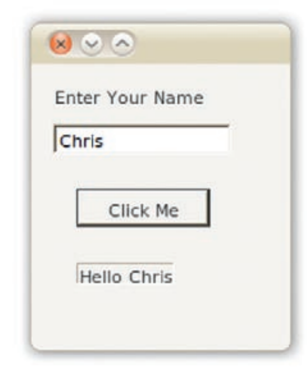
Проводим сравнения

С тех пор, как я в последний раз интересовался Mono, прошло несколько лет; тогда компилятор C# работал, но все средства развертывания были на базе командной строки, библиотека классов была ограниченной, и казалось, что далеко от консольного приложения «Hello World» здесь не уйдешь. Времена явно изменились, »



› Сборки – центральная идея .NET. Они являются «модулями развертывания» приложений .NET.





› Рис. 1. Найдите отличие! Простое приложение .NET Winforms, запущенное в Windows (слева) и в Linux (справа).

Мартышкина возня

Mono – по-испански «обезьяна» (отсюда и выбор логотипа). Его создали Мигель де Икаса [Miguel de Icaza] и Нат Фридман [Nat Friedman], которые тогда (на рубеже тысячелетия) работали в Ximian, компании, впоследствии приобретенной Novell. Ximian – другой вариант написания слова «simian» (обезьяны), и на логотипе самой компании изображен силуэт обезьяны. Это также объясняет название Bonobo (вид крупных обезьян) и логотип Gnome в виде обезьяньего следа. Пожалуй, назову-ка я очередной мой шедевр в .NET «Шерстистый гиббон».

и в этой статье я попытался дать ответ на два общих вопроса. Первый – выдерживает ли среда Mono сравнение со средой .NET от Microsoft? Например, пригодны ли сборки, созданные в Windows, для запуска в Linux? Второй – чем отличается опыт разработчика .NET в Windows и Linux? Иначе говоря, чем отличаются *MonoDevelop* и *Visual Studio*? Я сознательно ушел от обсуждения любых идеологических вопросов насчет разработчиков свободного ПО, пользующихся технологией Microsoft, маркетинговых стратегий Microsoft и опасений о патентах, связанных с .NET.

Сначала я понадеялся воспользоваться SLED или OpenSUSE: как-никак, Novell – спонсор Mono, и можно ожидать хорошей поддержки собственным брендом Linux. Увы, ссылка на OpenSUSE на сайте загрузки *MonoDevelop* оказалась нерабочей, а при попытке установить его на SLED возникла странная неразрешенная зависимость (из-за *Glibc*, ни больше ни меньше!). Итак, я отобрал четыре фанта у Novell и отправился на сайт (http://badgerports.org), где сидят последние версии пакетов Mono для Ubuntu 10.04, и они прекрасно установились. Для вашего сведения, я воспользовался Mono 2.6.7 и *MonoDevelop 2.4*.

> «Обещание Mono “пиши один раз, запускай везде” раздразнивает.»

Среды времени выполнения

Я решил начать с малого и убедиться, что в Linux запускается простое приложение **Windows Forms**, скомпилированное в Windows. Эта маленькая программа – мой эквивалент «Hello World» для Winforms. Нужно было просто скопировать сборку .NET (EXE-файл) из Windows в Linux и запустить его в Mono. Она запустилась успешно, как показано на рисунке (см. **рис. 1** вверху). Проясним происходящее: это не просто совместимость на уровне исходного кода – я взял приложение, скомпилированное в Windows, и запускаю его в Linux.

Работает ли это для более сложных приложений? Этот вопрос сводится к следующему: «Сколько классов среды .NET от Microsoft корректно реализованы в Mono?» Оцените масштаб: в библиотеке классов среды .NET (как мне говорили) около 11 000 классов. А поскольку в большинстве из этих классов по 20 или более методов и свойств, это *очень* много.

На сайте проекта Mono, www.mono-project.com/Compatibility, подробно представлены возможности .NET, реализованные и не реализованные в Mono. Итог, приведенный на этой странице, гласит: «Простейший способ описать все, что на данный момент поддерживает Mono, таков: все в .NET 4.0, кроме WPF, EntityFramework и WF; ограниченная поддержка WCF». Особо отметим, что поддержка Windows Presentation Foundation (WPF) не планируется. Гораздо более подробная информация приведена на Mono Class Status Pages, сайт http://go-mono.com/status. Здесь в браузере объектов мы можем продвигаться от пространств имен .NET к классам, методам и свойствам и видеть, что именно реализовано в Mono и что нет. Я заглянул в пространство имен **System.Drawing** (экранный снимок на **рис. 2** внизу) и обнаружил, что многие классы, такие как **Bitmap**, **Brush** и **Graphics** (выделены зеленым), полностью реализованы, но у класса **BufferedGraphics** есть метод (**Render**) с одним перегруженным вариантом, который нельзя реализовать в библиотеке *libgdiplus*.

Группа поддержки

Вооруженный этой информацией (и с учетом того, что на снимке показано лишь около 10 % одного из более чем 50 пространств имен) разработчик в принципе мог бы просмотреть свой код и понять, есть ли в нем что-то неподдерживаемое Mono. Если вы считаете, что это тупая задача, которую обязан делать компьютер, вы правы! В самом деле, утилита *Moma* (Mono Migration Analyzer) сделает это за вас. На входе *Moma* принимает набор сборок (файлов EXE и DLL), образующих приложение, проверяет их и сообщает, есть ли в них нереализованные функции.

Если *Moma* дает приложению «зеленый свет», это не гарантия его корректного запуска в Linux. У разработчика много способов определить по коду, на какой платформе он должен запускаться. Например, программа может искать внешние файлы в каталогах Windows или предполагать, что запущен локальный сервер IIS. *Moma*, безусловно, удобное средство для выявления проблемных областей, и это, конечно, приложение .NET.

Для ответа на вопрос, жизнеспособен ли Mono как инструмент для создания настоящих программ, совместимых с Windows и Linux, все эти детали экстраполировать трудно. Инстинкт шепчет мне, что это достижимо, если разработчик думает о совместимости с самого начала, но приложение, изначально написанное

mono — Mono System.Drawing in 4.0 vs MS.NET 4.0

Assembly **System.Drawing** last updated: 9 hours ago
System.Drawing.dll ×:8 :108 :62
System.Drawing ×:4 :61 :44
IDeviceContext
Bitmap
Brush
Brushes
BufferedGraphics :1 :1
System.IDisposable
void Dispose()
void Render()
void Render(Graphics target)
void Render(IntPtr targetDC) :1 :1
TODO Comment: The targetDC parameter has no equivalent in libgdiplus.
Graphics
BufferedGraphicsContext :1 :1
BufferedGraphicsManager

Legend
Missing: we need it
Extra: not present in MS.NET
Error: declaration mismatch
TODO: some functionality might be missing
The method just throws a NotImplementedException
Items colored in green are complete

› Рис. 2. На странице http://go-mono.com/status приведен детальный обзор: что реализовано в библиотеках классов Mono, а что нет.



для Windows без оглядки на совместимость, с гораздо меньшей вероятностью запустится в Linux без изменения исходного кода, т.е. без этапа портирования.

Среды разработки

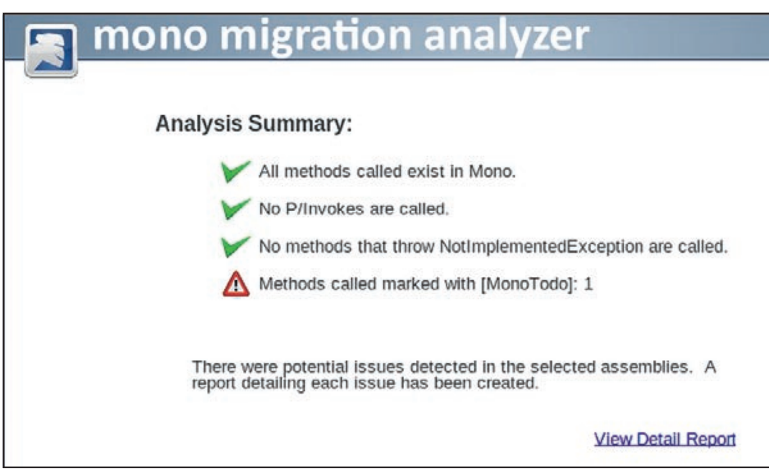
Теперь перейдем к сравнению сред разработки. Здесь менее понятно, что с чем сравнить. Microsoft предлагает набор средств разработки с постоянно расширяющимися возможностями (по постоянно растущим ценам), от *Visual C# Express* (бесплатного) до *Visual Studio Professional*, *Premium* и *Ultimate*. Наверное, лучше всего сравнивать с *Express* – все остальное только подчеркнет разницу.

Основное средство разработки в Mono – *MonoDevelop*. С *MonoDevelop* я столкнулся впервые и был приятно удивлен, увидев безупречную среду разработки с полным набором функций. В ней есть разворачиваемый текстовый редактор с подцветкой синтаксиса и обширной контекстной справкой, подобной IntelliSense от Microsoft. Имеются шаблоны кода (Microsoft называет их «snippets») – например, пустой цикл **for**, вставляемый в код парой щелчков мыши. Есть и встроенный отладчик – в нем можно использовать точки останова по условию, как в *Visual Studio*. Аналогичен и редактор свойств. Хотя между этими редакторами встречаются значительные различия, переход с *Visual Studio* на *MonoDevelop*, по-моему, должен пройти достаточно безболезненно.

Я попытался создать простую форму и был ошеломлен тем, что она основана на инструментарии *GTK*. Я ожидал увидеть архитектуру и набор панелей инструментов, аналогичные **Windows Forms**. Оглядываясь назад, скажу: это предположение было наивным – но помните, что я переходил на эту среду, привыкнув к *Visual Studio*/Winforms. Структура форм в *GTK* значительно отличается: гораздо большая роль уделяется «компонентам-контейнерам», определяющим раскладку, тогда как в Windows компоненты обычно размещаются по своим абсолютным координатам. (Подробнее об этом см. на www.mono-project.com/GtkSharp:_Widget_Layout_and_Packing).

Вообще-то существует конструктор форм Winforms для Mono (см. www.mono-project.com/WinForms_Designer), под названием *mwf-designer*. Я загрузил его и немного с ним поработал, но скоро согласился с комментарием на его домашней странице: «конструктор еще не готов для полноценной работы». Простую форму создать можно, но масса возможностей еще не реализована. Также нет встроенной возможности сборки/запуска/отладки, и код формы нужно сохранять в файл, а затем компилировать с командной строки (или, скорее, загрузить в конструктор *Visual Studio* – но я этого не пробовал). Звездный час этого конструктора, безусловно, еще впереди.

В Mono также есть набор утилит командной строки, похожих на предлагаемые Microsoft SDK. Есть, например, *gacutil*, утилита для работы с глобальным кэшем сборок (Global Assembly Cache – GAC), где находятся все разделяемые сборки. Также есть *wsdl* (утилита для написания прокси web-сервисов) и *monodies* (утилита для «разборки» сборок и просмотра их содержимого на промежуточном языке), с функциональностью, подобной *ildasm* от Microsoft.



› ***Moma*** **сигнализирует о возможной проблеме с запуском моего приложения .NET в Linux. Щелчок по ссылке «Просмотреть подробный отчет» точно определит сомнительный метод.**

Mono в мейнстриме

На разработку **Mono явно потратила кучу времени целая когорта умников.** Согласно сайту проекта, Mono запускается в Android, BSD, iOS, Linux, Mac OS X, Windows, Solaris и Unix, а также в ОС ряда игровых консолей, например, PlayStation 3, Wii и Xbox 360. Сам я проверил только Linux. Обещание «пиши один раз, запускай везде» очень раздразнивает аппетит, но Mono всегда будет играть в догонялки с Microsoft, которая продолжает рационализировать и вводить новые технологии в .NET, и трудно сказать, будут ли производители ПО ориентировать свои продукты на запуск в любой ОС с поддержкой .NET.

Пожалуй, еще рано оценивать значение .NET как независимой от платформы среды и Mono как ее полной реализации. Время покажет. LXF

Запуск программ Mono

В отличие от скомпилированных исходников или скриптов оболочки, программы .NET нельзя запускать, указывая только имя программы в качестве команды. Программу должен явно загрузить Mono, по команде

```
$ mono mydotnetapp.exe
```

Простое решение этой проблемы – создать маленькую обертку в виде скрипта оболочки вокруг каждого приложения .NET. Вот как это делается для утилит в дистрибутиве Mono. Пусть утилита *gacutil* находится в файле **/usr/lib/mono/2.0/gacutil.exe**, а ее скрипт оболочки – в **/usr/bin/gacutil**; в нем всего одна строка:

```
exec /usr/bin/mono $MONO_OPTIONS /usr/lib/mono/2.0/gacutil.exe
"$@"
```

Учтите, что *binfmt*, модуль ядра Linux, позволяет регистрировать форматы исполняемых файлов, распознавать их и передавать приложениям пользователей.

Учебники

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул *PostgreSQL* на защите кандидатской диссертации по физике элементарных частиц.

Продано

Чертовы буржуи!
Харухи

Жили-были эстонские программисты. Собрались они в организованную группу и написали P2P-клиент для обмена файлами *KaZaA*, не забыв добавить в него электронных шпионов. Но сказка не об этом, а о том, что те же эстонские программисты под предводительством шведа и датчанина на основе имеющихся наработок по распределенному трафику создали *Skype*. Не были они в своем начинании первыми, и не остались последними, но были самыми удобными в начале и самыми привычными впоследствии. Помогли *Skype* и недостатки Интернета, где NAT сейчас неизбежная реальность, а IP6 – прекрасное далеко.

Много людей пошло по пути установки *Skype* и ничего другого не хотят и видеть. Благо эта программа работает почти везде, в том числе и под Linux, включая Android. Но продолжится это всеобщее благо вряд ли. А почему? Да потому, что *Skype* – не свободная программа, и потому, что *Skype* на корню скуплен большой и мягкой корпорацией, которая последовательноне понимает, как работать со свободной инфраструктурой. Безусловно, при этом говорятся правильные и красивые слова о том, что ничего не изменится, а станет много лучше. Но очевидно, что *прямо сейчас* надо искать замену произведению эстонских программистов – и лучше свободную, чтобы не получилось «как всегда».

P.S. Вновь открылся сайт российского сообщества Creative Commons –

В этом месяце вы научитесь...

Совет месяца: История оболочки



Всем известно, что можно вызвать предыдущую команду, введенную в оболочке, нажав клавишу «вверх», но свойство истории большинства оболочек этим не исчерпывается.

Очень полезна возможность поиска в истории: нажмите Ctrl+R и наберите часть запущенной ранее команды. По мере ввода символов, оболочка будет показывать самую недавнюю команду, содержащую напечатанное – причем не обязательно имя команды: в истории ищутся совпадения в любой части строки.

Желая найти предыдущее совпадение, снова нажмите **Ctrl+R – при каждом нажатии появится сов**падение на шаг раньше. Теперь жмите Enter, чтобы выполнить команду еще раз, или курсор влево/вправо, чтобы начать редактирование.

С историей оболочки можно много чего понаделать. **!!** запускает последнюю команду еще раз. Почему удобнее делать это вместо курсора вверх? Причина такая: пусть вы только что ввели

```
apt-get install длинный-список-пакетов
```

и увидели ошибку, потому что у вас отсутствуют права root. Дело можно быстро поправить, выполнив команду

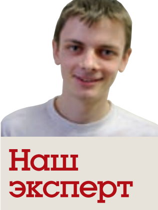
```
sudo !!
```

Это применимо и в скриптах оболочки, где команда строится динамически, и вы ее не знаете, пока она не сработает; но можно запустить ее снова через **!!**. Можно даже указать число команд, на которые надо вернуться: **!-2** запускает предпоследнюю команду, **!-3** – пред-предпоследнюю.

Email: Почтовый

Позвольте **Бобу Моссу** вырвать вашу почту из когтей Google-подобных сетевиков. Все ваши сообщения сольет воедино легкий в настройке клиент.





Наш эксперт

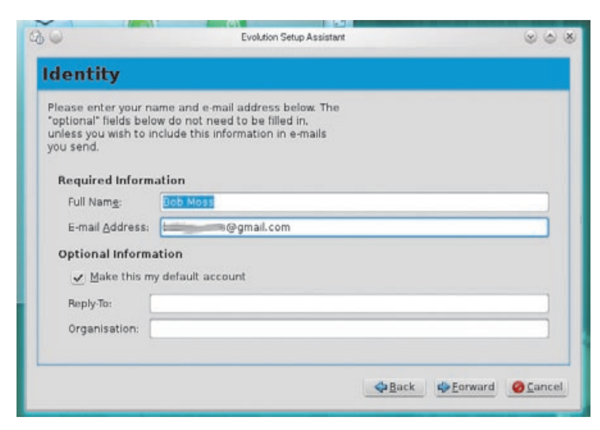
Боб Мосс
Разрывается между написанием кода и учебой по специальности «Компьютерные технологии».

Вполне простительно думать, что почтовый клиент уже умер и все ушли к поставщикам услуг сетевой почты; но, поразмыслив о безопасности и конфиденциальности наших данных в руках компаний, которым мы вынуждены доверяться, понимаешь, что эти умные пакеты рано списывать в утиль. Кроме того, как с любым интернет-сервисом, вы рискуете оказаться в ситуации, когда сервер недоступен или нет подключения к Интернету – а в результате нельзя и работать с почтой. Почтовые клиенты обеспечивают элегантное решение подобных проблем.

Опять же простительно думать, что почтовых клиентов вы сроду не видали. Однако вы обнаружите, что все основные дистрибутивы Linux, а также все версии Windows с середины 90-х годов, включали почтовый клиент в той или иной форме. В прошивку всех современных смартфонов и планшетов он также встроен, и компании, как правило, используют Microsoft Outlook или его аналоги для внутренней электронной почты.

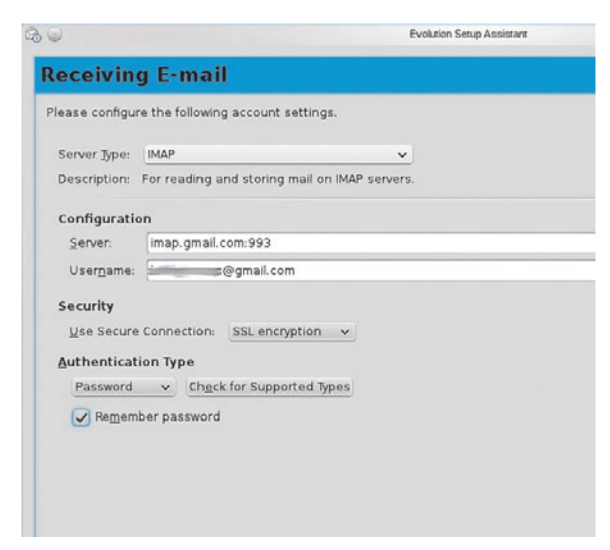
Управление почтой

Однако почтовые клиенты – вовсе не уцелевшие пережитки прошлого, подобные Брюсу Форсайту [старейший диктор британского ТВ, – *прим. ред.*] или институту монархии; они не менее актуальны сегодня, чем были когда-то. Где еще вы найдете решение, которое синхронизирует учетные записи электронной почты по работе, учебе и личные, напоминает, что завтра вы в 10 часов идете к стоматологу, и сообщает вам номер мобильника друга спустя два года после его получения? В довершение – если отсутствует связь с Интернет, вы не теряете доступ к почте с любой из ваших учетных записей: можно просто работать как обычно и синхронизировать все, когда вы снова будете онлайн.



› Рис. 1. Начните с добавки своего имени и адреса электронной почты.

По каким бы то ни было причинам полезности почтовых клиентов, мы начнем с *Evolution* (поставляемого по умолчанию с рабочим столом Gnome), а затем бегло осмотрим *Mozilla Thunderbird*. Прежде чем в омерзении выплевывать содержимое своих чашек какао, пользователи KDE будут рады узнать, что сразу же после мы охватим все, что можно извлечь из этих почтовых клиентов, используя достопочтенный менеджер персональной информации *Kontact*.

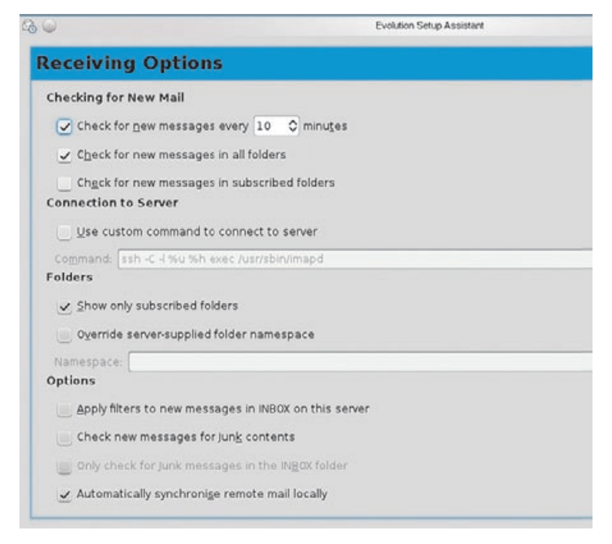


› Рис. 2. Можно выбрать из набора протоколов, таких как POP, IMAP, Novell GroupWise и Microsoft Exchange.

» **Месяц назад** Установили и настроили свой классный рабочий стол KDE.

Клиент в деле





› Рис. 3. Здесь устанавливаются интервалы проверки почты, выбирается, сканировать ли почту на содержание спама, и прочее.

Начнем с самой важной части установки – без нее серверы Google будут воротить нос от всех наших попыток связаться с ними. Войдите в http://gmail.com, затем в меню Настройки > Пересылка и POP/IMAP. Установите флажок Включить IMAP и нажмите кнопку Сохранить изменения.

Следующий шаг – «убить» окно вашего браузера и запустить *Evolution*. Пользователи Gnome обнаружат его в части Office или Интернет своего меню Приложения, а быстрый поиск по термину “Evolution” выручит тех, кто пользуется KDE-дистрибутивами или LinuxMint.

Следуйте инструкциям мастера

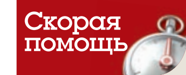
Первое окно, которое вы увидите (если вы раньше не запускали *Evolution*) – это мастер установки. Первые два раздела можно пропустить, нажав кнопку Далее. Появившееся окно должно выглядеть как **рис. 1**; добавьте здесь свое имя и полный адрес Gmail. Нажмите кнопку Вперед для продолжения.

Теперь вы должны получить окно, схожее с **рис. 2**. Вам надо выбрать IMAP в выпадающем меню, а затем установить входящий адрес сервера как imap.gmail.com: 993. Что это значит? Сервер, который *Evolution* будет опрашивать для почты – imap.gmail.com, а 993 – номер порта. Заметим, что это не работает с другими почтовыми клиентами, используемыми на нашем уроке, то есть придется установить номер порта потом.

В поле Имя пользователя добавьте свое полное имя пользователя, приплюсовав @gmail.com. Выберите также SSL и опции пароля в двух раскрывающихся меню, и установите флажок, чтобы сохранился пароль, который вы введете позже. Если вы любитель задешево получать острые ощущения, барабаня наугад по кнопкам, потеха вам обеспечена: нажатие кнопки Проверить поддерживаемые типы отобразит виды аутентификации, принятые у серверов Google! В рамках данного сценария это не столь важно, но поинтересоваться ими стоит, если вам нужно будет забирать электронную почту с других учетных записей (например, с вашего места работы) на более позднем этапе. Ну и еще раз нажмите Вперед, для перехода к следующему этапу работы мастера.

Настройка сервера

Теперь вы попадете в окно на **рис. 3**. Здесь важны самый верхний и самый нижний параметры. Верхний устанавливает интервал между проверками наличия новых сообщений, которые *Evolution* будет выполнять на сервере (10-минутный по умолчанию вполне разумен, но его можно изменить согласно вашим обстоятельствам). Нижняя опция *Evolution* позволяет хранить копии удаленных сообщений локально, а не таскать их заново при каждом опросе сервера, когда вы проверяете наличие новой почты. Так вы сэкономите полосу пропускания и получите постоянный доступ к вашей почте оффлайн.

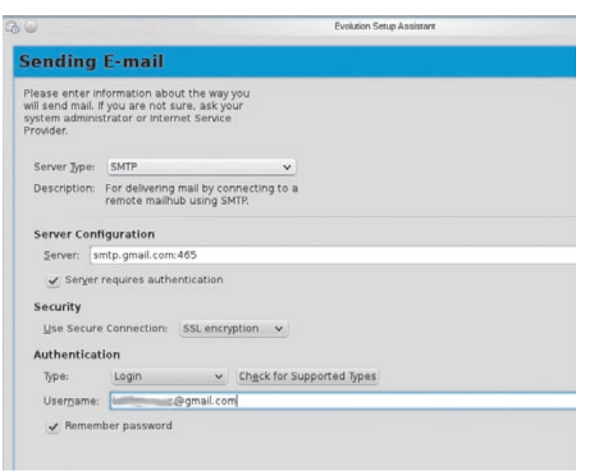


Для экспорта контактов из своей учетной записи Gmail нажмите на ссылку Контакты в верхнем левом углу страницы, затем в Дополнительные действия выберите пункт Экспорт... (сохраните как vCard для *Kontact* и *Evolution*, и выберите CSV, если у вас *Thunderbird*).

Чистим хвосты

После установки первым появится небольшое окошко ввода пароля. Здесь требуется ваш обычный пароль Gmail, и нажав кнопку OK, вы увидите папку «Входящие» в созвездии других папок (если вы не пользуетесь ярлыками).

Но идеал еще не достигнут. Если вы посылаете почту, бросаете письма в корзину или отмечаете их как спам, система пока что сохраняет их локально. Элегантным решением для таких сообщений будет переместить их в должное место вашей учетной записи Gmail. Для тонкой настройки папок Отправленные, Черновики и Удаленные перейдите в Редактировать > Настройки, выберите Gmail из списка учетных записей, нажмите кнопку Изменить, а затем выберите вкладку По умолчанию. А на ней просто выбирайте соответствующую папку из директории **[Google Mail]**.



› Рис. 4. Настройки по умолчанию для SMTP обычно разумны, но их легко и переделать.

Переместите свою почту

Вы будете рады узнать, что ваши почтовые клиенты имеют опции по перемещению почты, и при желании вы можете прыгать с одного на другой – или, как минимум, собрать электронную почту и настройки, а затем перемещать их по машинам.

Здесь мы рассмотрим причуды и слабости каждого пакета, о которых вам нужно знать загодя.

Evolution

Миграция между двумя *Evolution* проходит совершенно безболезненно. Просто войдите в меню Файл > Настройки резервного копирования для создания архива ваших писем и настроек. Теперь скопируйте пакет на любую другую рабочую машину, перейдите в меню Файл > Восстановление настройки – и вуаля! у вас есть две установки *Evolution* с одинаковыми сообщениями и настройками.

Переезд на другие почтовые клиенты немного сложнее, но отнюдь не невозможен. Например, чтобы переместить папку Входящие, выберите все сообщения в папке Входящие при помощи Ctrl+A, затем перейдите в меню Файл > Сохранить и задайте имя конечного файла **Inbox.mbx**. Это легко импортируется в большинство почтовых клиентов (от *Microsoft Outlook* до перечисленных здесь примеров).

Следующий этап портирования, естественно, экспорт контактов и календарей. Первое достигается щелчком правой кнопки мыши по адресной книге под разделом «Контакты» и выбором Сохранить как vCard, а второе можно осуществить, нажав правой кнопкой мыши на календарь и выбрав Сохранить на Диск. Форматы vCard и iCal оба совместимы почти со всеми аналогичными приложениями.

KMail

Если вы переходите из другого почтового клиента на *KMail* на той же машине, вас обрадует, что этот пакет облегчает жизнь. Просто перейдите в меню Файл > Импорт сообщений и выберите вашу установленную программу в выпадающем меню. *KMail* импортирует все ваши письма и структуры папок как они есть. Если вы переходите с другого компьютера, можете использовать тот же инструмент для импорта почтовых ящиков, ранее экспортированных в файл с расширением**.mbox**. Для экспорта почты из *KMail* применим тот же трюк, что и в *Evolution* – нажмите Ctrl+A для каждого почтового ящика и сохраните конечный файл в формате MBOX, чтобы запросто перемещать вашу почту в другие пакеты.

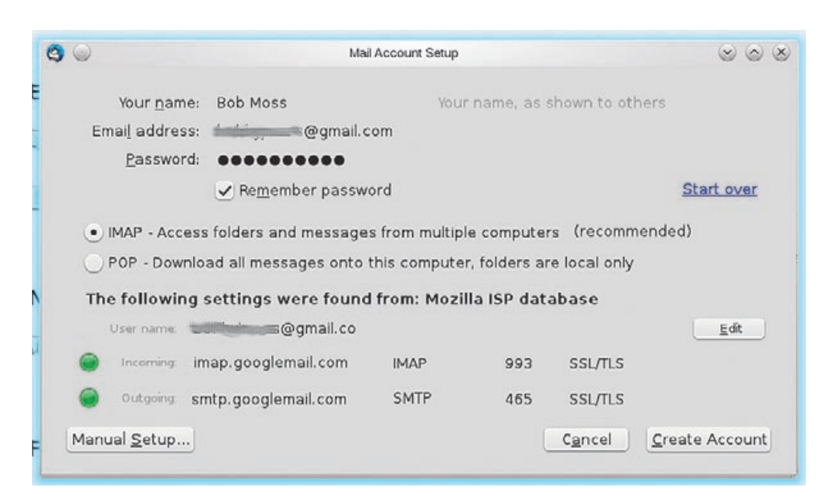
KMail – специализированное приложение, и ориентирован исключительно на обработку почты, чего и следовало ожидать от программы, которая интегрируется с другими инструментами в рабочем столе KDE. Это основная причина, почему (как и в *Mozilla Thunderbird*) здесь не хватает управления контактами и поддержки календарей. Однако мириады опций импорта почты из ряда других приложений и источников делают *KMail* достойной альтернативой.

Mozilla Thunderbird

К сожалению, *Mozilla Thunderbird* пока не способен на несложный импорт или экспорт. Хотя адресные книги импортировать можно(если вы сохраните их в формате CSV, а не vCard), почту *Thunderbird* умеет извлекать только из *Netscape Communicator* с помощью Сервис > Импорт... – но ведь это ПО каменного века, которым по жизни никто не пользуется. И не странно ли, что порт Windows признает другие почтовые клиенты, а порт Linux – нет.

Наш совет таков: если вы не против дополнительных настроек вручную и перемещение почты для вас играет ключевую роль, берите какой-нибудь другой из почтовых клиентов нашего урока. А если вам надо просто переводить почтовый клиент при каждом переходе на новую машину (или вы решили запустить портативную версию *Mozilla Thunderbird*), *Thunderbird* по-прежнему стоит иметь в виду.

Наконец, в следующем окне (**рис. 4**) задайте параметры передачи. В верхнем выпадающем меню выберите SMTP – это протокол сервера, который вы будете использовать. Адрес сервера исходящей почты – smtp.gmail.com:465 (как и в предыдущем IMAP-адресе, smtp.gmail.com – адрес сервера, а 465 – номер порта). Снова введите полный адрес Gmail и, в зависимости от вашей конфигурации, выберите Простые или Авторизацию по имени пользователя. (Последнее лучше, так как гарантирует шифрование вашей почты, но работает не на всех установках – по причинам, которые никто адекватно не объяснил. При неполадках с отправкой электронной почты рекомендуется вернуться в Простые.) Заключительный этап после этого – призвать учетную запись Gmail в следующем окне и нажать кнопку Готово. Затем нажмите Применить в последнем окне и дождитесь перезагрузки *Evolution*.



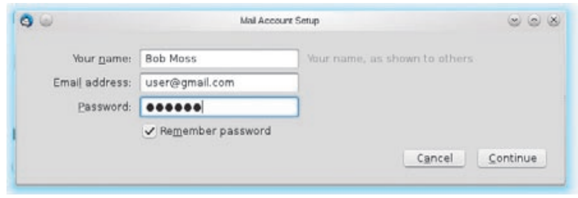
› **Если введенная информация верна, переходите к следующему шагу процесса.**

Раньше в процессе настройки учетной записи Google Mail в *Mozilla Thunderbird* для начальной установки надо было пройти несколько этапов: ваш логин, затем параметры входящей почты, а затем – настройка SMTP-сервера. С тех пор Mozilla сделала клиент непритязательно простым в настройке получения и отправки почты через множество сервисов web-почты и Интернет-провайдеров.

Настройка Thunderbird

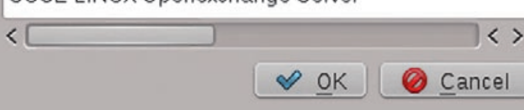
Если вы запустили *Thunderbird* впервые, введите свой адрес и пароль Gmail (как на **рис. 1**). *Thunderbird* получит требуемые настройки для учетной записи Gmail с серверов Mozilla и спросит ваше подтверждение. Мы намеревались использовать IMAP на порте 993 через imap.gmail.com, а для отправки – SMTP через порт 465 на smtp.gmail.com, и это, скорее всего, будет корректно отражено в окне (см. **рис. 2**) – просто нажмите Далее, и все создастся автоматически.

Вот бы жизнь всегда была такой простой! Увы, это не так, и если ваша машина находится за прокси-сервером или необходимо устранить проблемы с подключением, целесообразно будет попробовать делать это вручную. Перейдите к Правка > Параметры учетной записи... и в появившемся окне выберите Действия с учетной записью, внизу справа. Теперь выберите учетную запись



› **Введите по запросу свое имя пользователя, а *Thunderbird* доделает остальное!**

» **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.



› **Настройте свои календари на обновление, как и вашу почту.**

Жизнь полна событий

Итак, почта синхронизируется локально; почему бы не расширить эту функцию, включив ежедневные задачи? Если вы используете Google Calendar, можно и его настроить на автоматическую синхронизацию с *Evolution*. Для этого нужно начать новый календарь, но вместо выбора Google из выпадающего меню (что более интуитивно), выберите В Интернет. Альтернативно, если ваш дистрибутив позволяет, можно выбрать вариант Google из того же меню и нажать кнопку Получить список, чтобы не пришлось копировать и вставлять ссылки.

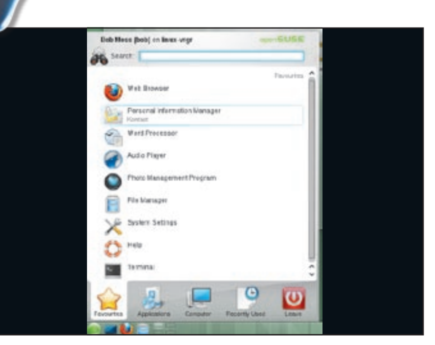
Если вы не из везунчиков, потребуется открыть браузер и перейти в службу Календарь Google; нажмите кнопку раскрывающегося значка рядом с календарем, который хотите импортировать, и выберите Настройки календаря. Рядом с Личным Адресом, скопируйте ссылку, содержащуюся в iCal, и вставьте эту информацию в разделе URL-адрес.

Назвать календарь можно как угодно, но не забудьте использовать полный адрес Gmail в качестве имени пользователя. При желании можно пометить его как календарь по умолчанию, а затем скопировать его на местном уровне для использования в автономном режиме. Введите свой пароль, и календарь появится автоматически и может быть легко изменен.

Вы можете добавить любое количество календарей Google, и этот метод должен сработать с большинством других календарей онлайн-сервисов.

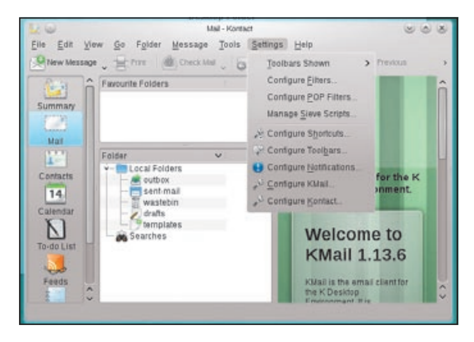
электронной почты. Во всплывающем окне (оно будет выглядеть, как на **рис. 1**) введите свой адрес электронной почты и пароль, как и раньше, но на сей раз, когда мастер попытается найти настройки учетной записи, выберите Настройка вручную... а не Далее. С этого момента вы увидите, что меню и диалоги аналогичны тем, которые вы ожидали бы в *Evolution*; это упрощает настройку. LXF

Шаг за шагом: Устанавливаем Kmail



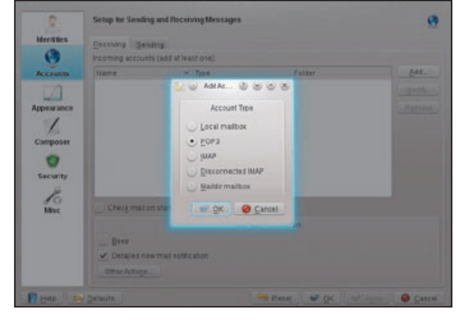
1 Запустите Kontact

«Правильный» способ – запуск менеджера персональной информации *Kontact* в разделе Избранное меню KDE.



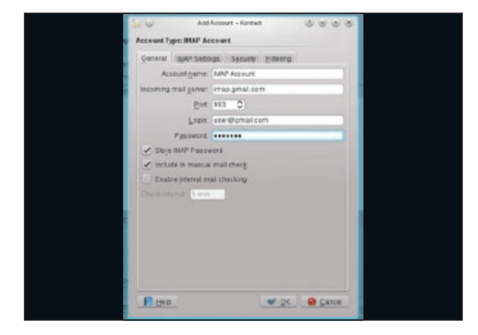
2 Запуск установки

Для начала настройки вашей почтовой программы, нажмите на иконке почты слева и перейдите в меню Настройки > Настроить Kmail...



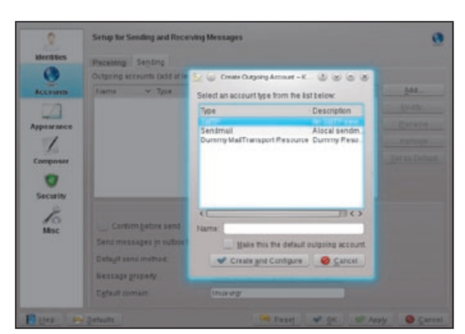
3 Добавьте учетные записи

Нажмите кнопку в разделе Учетные записи в левой панели, затем нажмите кнопку Добавить... в правой части окна.



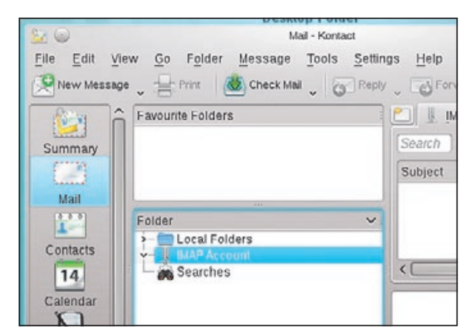
4 Добавьте детали

Далее необходимо заполнить некоторую информацию: выбрав IMAP, вставьте детали IMAP в появившемся окне.



5 Настройте отправку

По завершении можете нажать на вкладку Отправка и ввести информацию SMTP для вашей учетной записи Gmail.



6 Получить почту

Сделав это, нажмите кнопку Применить, а затем выберите Проверить почту из главного окна, чтобы начать использовать *KMail*.

» **Через месяц** Рассмотрим особенности ультра-легкого рабочего стола *Xfce*.

DreamPlug: Чудо-

Грэм Моррисон создает точку беспроводного доступа, NAS и музыкальный потоковый плейер из коробки не крупнее упаковки из-под гаванских сигар.





› **Съемный блок питания можно воткнуть прямо в розетку или через удлинитель.**



Наш эксперт

Грэм Моррисон
Если Грэм не размышляет, что бы еще запихнуть в **LXF**, и не возится со старыми синтезаторами, то, возможно, ищет способы повысить продуктивность в KDE 4.

Благодарим www.newit.co.uk за предоставление DreamPlug!

За последние пару лет произошло нечто вроде революции в комплектации компьютеров с Linux. На верхнем конце шкалы – сети суперкомпьютеров, постоянно побивающих мировые рекорды скорости вычислений, хотя прогнозы погоды точнее от этого не становятся. На другом конце – полнофункциональные компьютеры с Linux размером с обычный блок питания. По этой причине данный форм-фактор был назван вставными компьютерами-вилками [plug computing], а благодаря масштабируемости Linux свободное ПО придает этим блокам исключительную мощь.

Одним из самых популярных устройств стал SheevaPlug, небольшой белый блок с ARM-совместимым CPU, Ethernet и связью по USB. Он использовался везде: от мощных специализированных NAS-решений и домашних сетевых серверов до контроллера Arduino для заказного оборудования, «умного дома», световых индикаторов и обеспечения музейного киоска. Благодаря увязанности большинства этих устройств со специальной ARM-версией Ubuntu, и позднее Debian 5, вы могли устанавливать пакеты и настраивать службы совершенно так же, как на вашем домашнем настольном компьютере или сервере, то есть совсем не нужно было быть экспертом, чтобы система работала: получите доступ к их SSH-серверу из командной строки и принимайтесь за настройку.

Однако SheevaPlug был несколько ограничен по части подключений. Например, если он использовался как источник данных, данные обязаны были исходить из USB-порта. Эта проблема была решена новым устройством, GuruPlug, которое обзавелось слотом eSATA, MicroSD, да еще и Wi-Fi, и превратило вставное устройство в куда более жизнеспособную альтернативу обычному компьютеру Linux. Но и тогда оставалось место для усовершенствований – и вот появился DreamPlug.

Он представляет третье поколение вставных устройств с Linux и комплектуется со всем предполагаемым в подобном устройстве, с теми же возможностями подключения, что и предшественники, плюс bluetooth и беспроводное радио, быстрый CPU, вход для микрофона, выход для наушников и даже выход для цифрового аудио. Емкость корневой файловой системы 2 ГБ, не надо обращаться к внешним носителям, и это идеал для вставных компьютеров. Он будет работать и прикрепленным к задней крышке вашего усилителя или телевизора, и прикорнув в коробке из-под обуви или сигар. Нет, пожалуй, не из-под сигар: ему нужна хорошая вентиляция.

> «DreamPlug идеален для самых разных решений на базе Linux.»

Все это превращает DreamPlug в идеальный инструмент для самых разных решений на базе Linux, а мы возьмем его за основу для нескольких проектов класса компьютеров-вилок. Но это не значит, что вам нужен именно DreamPlug. С той же легкостью можно употребить более старое вставное устройство и даже старый нетбук. Большая часть наших инструкций годится и для них. Секретное оружие – не оборудование, а Linux, то есть эти инструкции можно адаптировать для любого устройства.

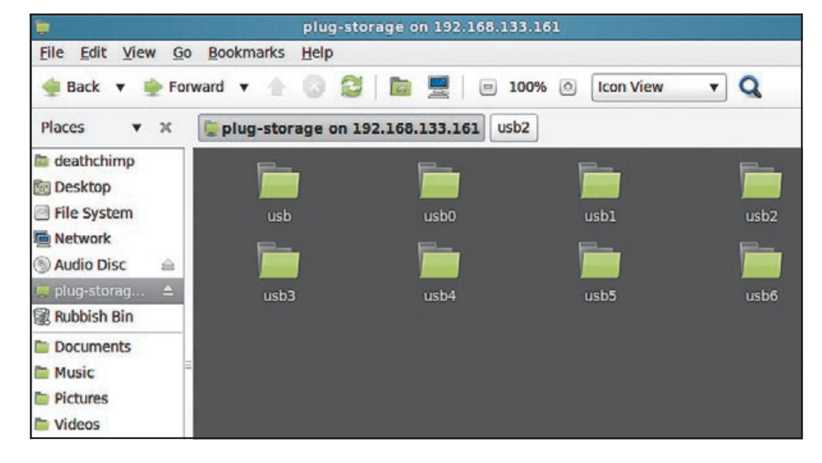
коробочка

Подключение к устройству

Приведение в рабочее состояние близко к реалиям "plug and play". Единственная потенциальная сложность – поиск IP-адреса для вашего устройства после его подсоединения к LAN и включения. Если вы дома и подключаетесь через обычный маршрутизатор, можно просмотреть подключенные устройства в web-интерфейсе маршрутизатора – IP-адрес компьютера-вилки обнаруживается легко. К нему можно обратиться, введя на компьютере с Linux команду **ssh root@ipaddress** и указав **nosoup4u** на запрос пароля. Управление вашим оборудованием не сложнее, чем выполнение простых команд, совсем как если бы вы сидели за терминалом вашей машины с Linux.

При нахождении в большой LAN дела могут осложниться, если вы не сумеете изолировать присоединение к вашему локальному компьютеру.

К счастью, DreamPlug преднастроен как точка доступа беспроводного соединения, а следовательно, простейший способ подключиться к устройству – через ноутбук посредством точки доступа **Plug2-uAP-504c**. Пароля не требуется, а если ваш DreamPlug подключен еще и к маршрутизатору по Ethernet, вы получите доступ в Интернет через беспроводное соединение. Однако открытая по умолчанию точка доступа создает серьезный риск для безопасности вашей LAN, так что следовало бы добавить пароль или вообще отключить точку доступа. В следующем разделе мы расскажем, как превратить ваше вставное устройство в удобную точку доступа для домашних нужд, вместе с двумя портами Ethernet для максимального потенциала маршрутизации.



› Даже без всякой настройки DreamPlug недостает только USB-накопителя, чтобы стать полнофункциональным сетевым хранилищем.

Настройка точки доступа

Подключившись к точке доступа, введите в командной строке компьютера с Linux **ipconfig**. Вы увидите, что беспроводная сеть использует адреса из диапазона 192.168.1.*, а также IP-адрес для соединения по Ethernet.

Поищите в выводе командной строки **Link encap:Ethernet** и **inet addr:** и воспользуйтесь этим IP-адресом, чтобы подключиться к вставному устройству, не обращаясь к его беспроводной сети. Дело в том, что мы собираемся поменять беспроводную конфигурацию, а при работе беспроводной сети этого сделать нельзя.

Первым делом, до всех других изменений, введите **apt-get update**. Это обновит все установленные пакеты до самой свежей версии.

Спецификация DreamPlug

DreamPlug немного больше, чем обычный блок питания, его размер 108 × 54 × 24 мм, однако мощность его CPU далеко не легковесная.

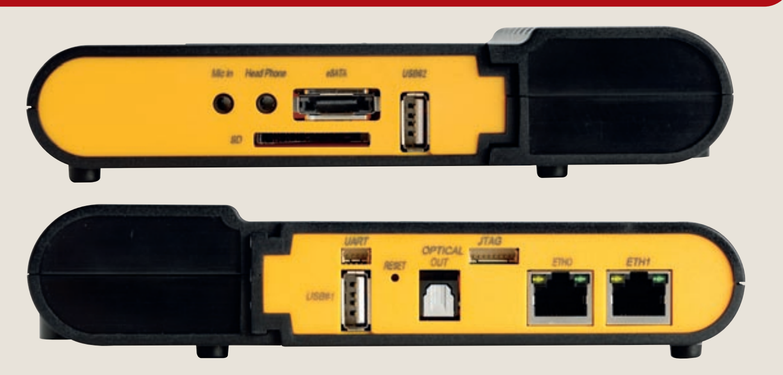
Там находится ARM-совместимое ядро 1,2-ГГц Marvell Sheeva и 512-МБ DDR2 ОЗУ – такая спецификация типична для пустых блоков NAS, которые дороже вдвое.

Файловая система установлена на встроенной 2-ГБ карте микро-SD, из которых в стандартной поставке системы используется 80 %, и можно заказать дополнительный модуль JTAG для перепрограммирования кода загрузки и отладки системы.

И, наконец, весь прибор потребляет ничтожные 5 Вт мощности, и в наших тестах был теплым, а не горячим, что весьма впечатляет, если учесть, что вентилятора нет.

Цена: £135

Сайт: www.newit.co.uk



› 2×ГБ Ethernet, 2×USB 2, 1×eSATA 2.0, 1×SD кард-ридер, Wi-Fi 802.11 b/g, Bluetooth BT2.1, выход для наушников, вход для микрофона, оптический аудиовыход SP/DIF.

Также следует заранее позаботиться о доступе к резервной копии образа системы: она вас выручит, если вы напортачите до неремонтопригодности. И вообще, делая изменения в любом файле настройки, обязательно сохраняйте рабочую копию. Все подготовив, поменяйте пароль root на нечто не публикуемое в популярных журналах, с помощью команды **passwd**.

Нам не потребуется устанавливать и настраивать слишком много, ведь устройство уже настроено как точка беспроводного доступа. Например, о DHCP-сервере хлопотать незачем: он уже установлен и работает. Остается поменять настройки безопасности – и вместо обычного способа, при помощи **ifconfig** и **wpa-supplicant**, для этого существует **uaputl**, собственная команда GuruPlug и DreamPlug, управляющая беспроводным оборудованием.

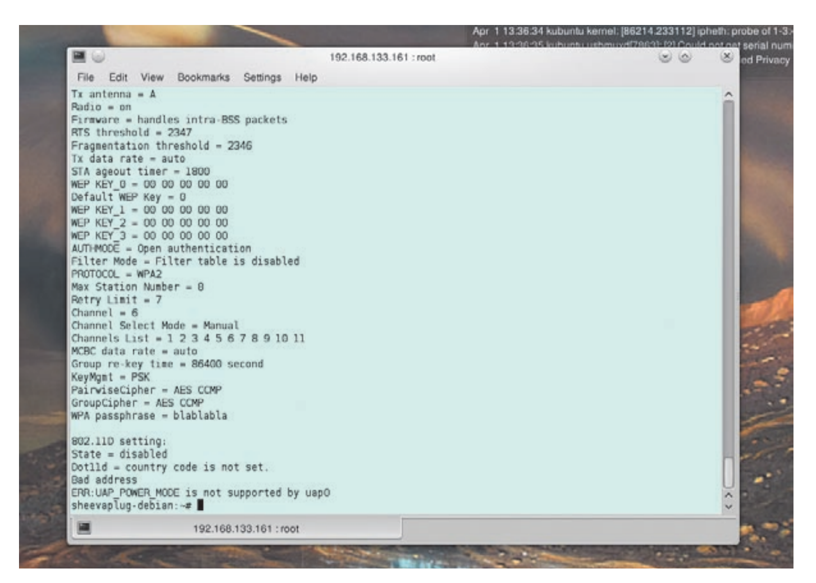
Например, введя в командной строке **uaputl sys_config**, вы увидите длинный список текущих настроек устройства, включая SSID и все настройки безопасности.

Также можно командой **uaputl** менять опции. Например, введя **uaputl sta_list**, вы увидите список всех компьютеров, подключенных на данный момент к беспроводной сети, а выполнив **uaputl bss_stop**, **uaputl sys_cfg_ ssid "MySSID"** и **uaputl bss_start**, можно изменить SSID. А если вы хотите усилить безопасность, то метод будет аналогичным:

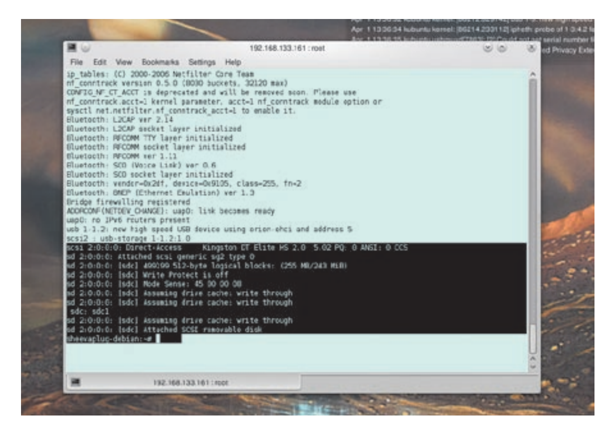
```
uaputl bss_stop
uaputl sys_cfg_protocol 32
uaputl sys_cfg_wpa_passphrase "password"
uaputl sys_cfg_cipher 8 8
uaputl bss_start
```

Если вам нужно запросить беспроводное устройство, какие именно протоколы и шифры оно поддерживает, просто выполните ту же команду, что и выше, но последний аргумент замените на **-help**. Это следует проделывать при каждом перезапуске вставного устройства, так что лучше будет изменить стартовый скрипт, используемый вставным устройством для настройки сети при каждой загрузке. Тогда вы будете защищены, даже если произойдет сбой питания и вставной компьютер перезапустится без вашего присмотра. Стартовый скрипт для сети находится в **/root/init_setup.sh**. Если открыть этот файл в редакторе командной строки **nano**

> **«Эти устройства прекрасно подходят как сетевое хранилище.»**



› Беспроводной модуль легко настроить и как стандартный сетевой клиент, а не только точку доступа.



› Подключаясь к новому устройству, введите 'dmesg' , чтобы увидеть вывод системного журнала и определить id устройства.

и добавить после строчки **/usr/bin/uaputl sys_cfg_ssid $SSID** три строчки с **sys_cfg** из примера выше, у вас будет безопасная система с самой загрузки, и отличная точка доступа для вашей сети.

Сетевое хранилище

Другая задача, для которой прекрасно подходят устройства такого типа – сетевое хранилище (NAS): благодаря простоте, с которой добавляются внешние приводы к порту eSATA, и подключениям, предоставляемым USB-портами и слотом SD-карты и идеальным для переноса музыки и изображений с внешнего устройства. В полнофункциональный NAS превратить DreamPlug также очень просто – ведь это тот же процесс, что проделывается на обычном компьютере с Linux.

Первый шаг – обеспечить распознавание хранилища устройством. Способ зависит от того, что вы подключаете. Например, диск eSATA или USB-брелок появляются в дереве устройств как дополнительный жесткий диск, и большинство устройств такого типа имеют горячее подключение – то есть допускают подсоединение при уже работающей системе. Вставив USB-брелок, введите через несколько секунд **dmesg**, и вы увидите примерно такой вывод:

```
sd 2:0:0:0: [sdc] 499199 512-byte logical blocks: (255 MB/243 MiB)
```

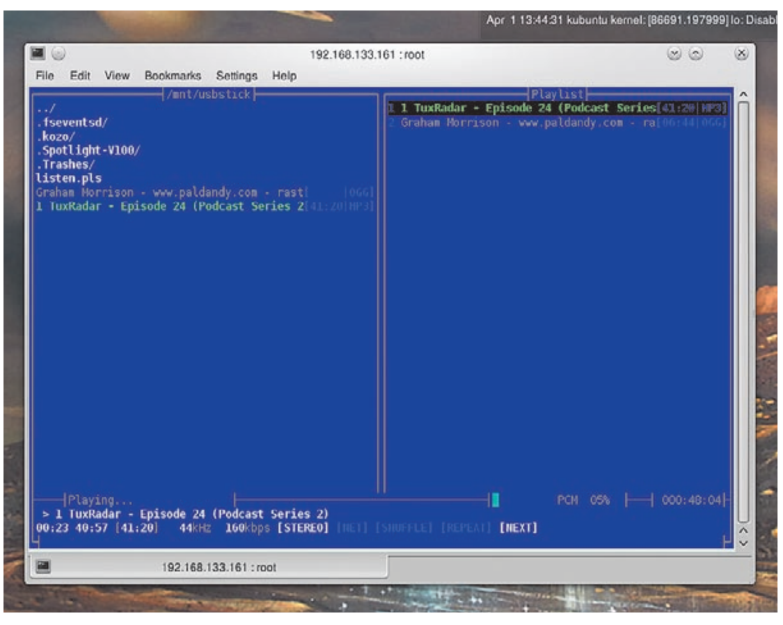
Он показывает, что привод распознался и подключен к файлу устройства **/dev/sdc**. Если вы уже сделали разбивку и форматирование на этом носителе, следующим шагом будет монтирование привода в файловую систему. Для этого создайте каталог монтирования в **/mnt** (**mkdir /mnt/usbstick**) и вызовите команду монтирования с двумя аргументами:

```
mount /dev/sdc1 /mnt/usbstick
```

Эта команда берет узел устройства из вывода **dmesg**, но первым разделом в устройстве **sdc** является **sdc1**, а не само устройство. Для большинства устройств файловая система автоматически определяется командой **mount**, поэтому мы не указали ее явно. Если при этом возникают проблемы – значит, у вас старая или экзотическая файловая система; обратитесь к опции **mount -t**.

В противном случае вы должны обнаружить, что содержимое внешнего привода расположено в папке **/mnt**, созданной как точка монтирования. Чтобы это делалось автоматически при загрузке системы, добавьте с помощью текстового редактора **nano** в файл **/etc/fstab** следующие строки, изменив тип файловой системы на используемый на вашем разделе:

» **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.



› **Декодирование MP3 в Mocp требует только 5% мощности процессора DreamPlug, остальное остается в вашем распоряжении.**

```
/dev/sdc1 /mnt/usbstick ext3 defaults 0 0
```

Сейчас у нас есть достаточно места для хранения любых файлов, которые будут обслуживаться NAS, и следующим шагом станет запуск сервера, чтобы эти файлы стали доступными.

Чаще всего для NAS используется сервер *Samba*, свободное приложение, которое заново реализует многие из сетевых протоколов Microsoft. В DreamPlug замечательно то, что *Samba* уже готов к работе.

В рабочем столе Gnome, просто нажмите в меню Places на **Connect to Server** и в появившемся окне выберите **Windows Share**.

Останется только ввести IP-адрес вашего вставного устройства, и вы увидите, как в сетевой папке появится **plug-storage**. Пользователи KDE могут проделать то же самое, открыв *Dolphin*, нажав на **Network** в панели слева, а затем на **Add Network Folder**. Выберите там сетевую папку Windows, введите имя для IP-адреса и **/** для папки, и после нажатия на **Finish** вы увидите ту же папку, что и пользователи Gnome.

Появившаяся папка **Plug-storage** является точкой доступа для самых разных устройств, подключаемых к DreamPlug, включая кард-ридер с корневой файловой системой и образами ядра, а также всех присоединенных устройств. Например, файлы с нашего USB-брелка можно отыскать в папке **usb2**. Это простое решение для обслуживания файлов, но, как и в случае с точкой доступа, которую мы настраивали ранее, сейчас любой может получить доступ к вставному устройству, что не есть хорошо. Решение состоит в изменении конфигурации *Samba*.

Файл настройки расположен в **/etc/samba/smb.conf**. Открыв его в текстовом редакторе **nano**, в конце файла, после заголовка **[plug-storage]**, вы найдете раздел для текущих совместных ресурсов, с тем же именем, что и общий ресурс. Рекомендуем первым делом сменить **path** с папки **/media** на тот, что вы создали для вашего подключенного устройства. Тогда нанесенный вашей системе вред ограничится вашими мультимедиа-файлами, а не самой системой; а если вам нужна дополнительная безопасность, установите writeable в **no**.

Чтобы эти изменения возымели действие, перезапустите сервер *Samba* командой **/etc/init.d/samba restart**. Теперь, подключаясь к серверу тем же способом, что использовался ранее, можно будет увидеть только файлы на внешнем носителе.

Однако ради еще большей безопасности следует приказать *Samba* делиться доступом только при указании дозволенного имени пользователя и его пароля. Это проделывается, во-первых, путем изменения в файле настройки *Samba* **public** на **no**, а затем добавлением пользователя *Samba*. Но сначала создайте базового системного пользователя. Запустите сначала команду **adduser username**, вводя детали по запросу. Затем введите **smbpasswrd -a username**, чтобы создать пользователя *Samba*. В обоих случаях задайте одно и то же имя пользователя. После очередного перезапуска сервера *Samba* и подключения к тому же самому общему ресурсу **plug-storage**, у вас запросят имя пользователя и пароль, и доступ к файлам будет только после ввода ваших полномочий *Samba*. Теперь передавайте полномочия всем, кому вы хотите дать доступ, или создайте для них отдельную учетную запись.

> «Замечательная штука — цифровой оптический аудиоразъем.»

Воспроизведение музыки

Еще одна замечательная штука в DreamPlug – цифровой оптический аудиоразъем. С его помощью можно передавать цифровой аудиопоток на любой стандартный усилитель с оптическим цифровым аудиовходом. А так как аудио остается в цифровом виде, не преобразовываясь в аналоговый сигнал, качество выхода будет исключительным. Даже если ваше устройство старое и не содержит аудио, его все равно очень легко подключить через стандартное аудиоустройство USB, большая часть которых распознается и настраивается не заставляя вас марать рук.

Звуковой интерфейс DreamPlug уже готов к работе в качестве устройства ALSA, что делает проект особо простым в обращении. Главной задачей является поиск, к чему его применить. А поскольку большинство из нас собирается использовать вставное устройство посредством командной строки, то искать будем музыкальные программы, работающие в командной строке. Например, многим нравится MPD – Music Player Daemon. Это простой сервер, который воспроизводит музыку, с поддержкой множества плагинов и разнообразных методов контроля. Возможно, это лучшее решение, но если вам нужно больше обратной связи, есть более простые варианты.

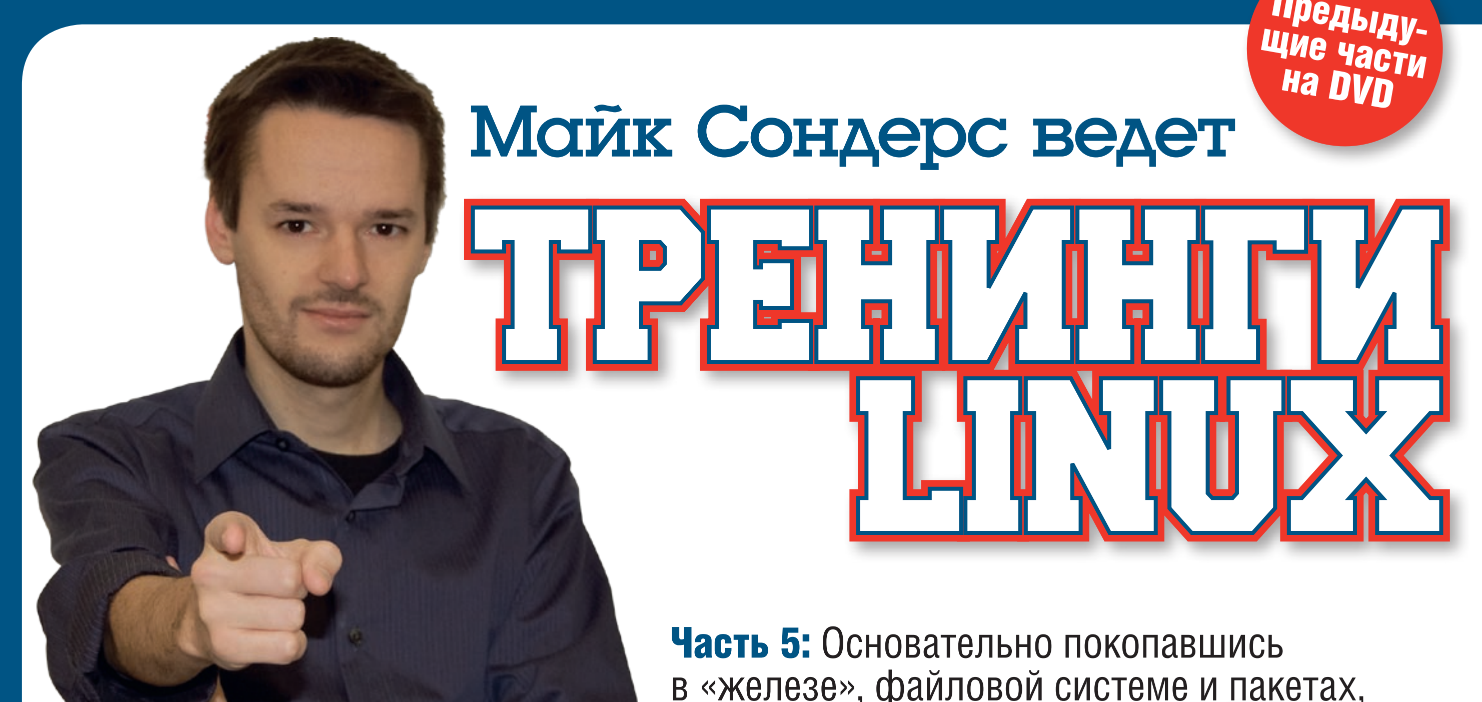
Менеджер пакетов Debian предлагает хорошую коллекцию утилит, предоставляющих полезные пользовательские интерфейсы, так что воспроизведение музыки в командной строке происходит как будто на рабочем столе. Наш любимый плейер называется *Mocp*, что, возможно, значит "music on the console player [музыка на консольном проигрывателе]". После установки командой **apt-get install mocp** рекомендуем сменить каталог на тот, где вы храните аудиофайлы, и запустить программу, введя **mocp -R alsa**.

После этого появится главный вид воспроизведения: слева – список локальных файлов, а справа – пока пустой плей-лист. Нажатие **h** выдаст сочетания клавиш для команд управления интерфейсом. Например, нажмите **a**, чтобы добавить файл в плей-лист, табуляцию – чтобы переключаться в вид плей-листа, а пробел – для начала воспроизведения. Пробелом также можно ставить паузу и возобновлять проигрыш, а громкость воспроизведения меняется клавишами **<** и **>**. Это 90 % команд, необходимых при использовании *Mocp,* а про остальные можно узнать, нажав на **h**.

Но самое лучшее мы приберегли на закуску. Нажмите **q** для возврата в командную строку. Вы должны заметить, что музыка не остановилась: она все еще играет в фоновом режиме. Это оттого, что для обработки музыки *Mocp* использует простой сервер, а GUIкомандной строки – это метод, используемый для добавления музыки в плей-лист. LXF



Если скачать в вашу папку с музыкой файл *Shoutcast* **pls**, то в командной строке можно прослушивать интернет-радио, в точности как файл Ogg или MP3.



Майк Сондерс ведет Тренинги Linux

Часть 5: Основательно покопавшись в «железе», файловой системе и пакетах, пора полностью освоить командную строку.

Mike



Наш эксперт

Майк Сондерс пишет о Linux более десяти лет, и установил больше дистрибутивов, чем съел горячих обедов.

Отдельные скептики готовы заставить вас поверить, что командная строка – замшелый пережиток семидесятых, бессмысленное развлечение умников, которое обычным людям не стоит и трогать. Но когда дело доходит до системного администрирования, большей ошибки нельзя и придумать. Командная строка, она же – оболочка, важна здесь как никогда – и по веским причинам:

» Она всегда с вами. Она существует за всеми слоями графики, видимыми в типичном настольном дистрибутиве Linux, и даже если оконный менеджер работает некорректно, можно нажать Ctrl+Alt+F2, переключиться в командную строку и исправить неполадки.

» Ей не нужна графика. На компьютер можно зайти удаленно (по SSH) с другого конца света по коммутируемому соединению и работать так же, как на локальной машине. Не нужны тормозные VNC или удаленный рабочий стол. Аналогично, на многих компьютерах, например, на серверах, графический интерфейс ни к чему. Достаточно командной строки.

» Она пряма и непосредственна. Она делает именно то, что ей велено. Никакой ерундистики вроде «кликните по красной кнопке недалеко от левого верхнего угла окна, затем найдите меню Foo и поставьте галочку рядом с ним». Вы просто набираете именно то, что должен выполнить компьютер, и он выполняет это. Никаких хождений вокруг да около.

Поэтому все хорошие администраторы обладают глубоким пониманием командной строки, и если вам нужен сертификат LPI, вы должны усвоить идеи и утилиты, о которых мы поговорим. Если вы новичок в Linux, это еще и прекрасный способ понять, насколько мощный и гибкий инструмент командная строка. В предыдущих статьях мы применяли несколько отдельных команд, а теперь изучим *Bash* – оболочку по умолчанию в 99,9 % дистрибутивов Linux – более глубоко; итак, начнем!

Раздел 1: Ориентируемся

Если у вас графический рабочий стол Linux, командную строку можно открыть с помощью меню – она обычно называется Terminal [терминал], Shell [оболочка], *XTerm* или *Konsole* [консоль]. В данном случае мы работаем в CentOS 5.5, где для вызова командной строки используется пункт меню **Applications** > **Accessories** > Terminal (Приложения > Стандартные > Терминал). После ее запуска мы увидим вот что:

```
[mike@localhost ~]$
```

Это строка приглашения, и она состоит из четырех частей. Первая – имя пользователя, который в данный момент находится в системе; здесь это **mike**. За ним следует имя хоста нашего компьютера – **localhost**. Тильда (**~**) показывает, в каком каталоге мы находимся; например, вместо нее могло бы быть **bin**, если бы мы находились в каталоге **/usr/bin**. Обычно сеанс терминала начинается в домашнем каталоге пользователя, и тильда – короткий способ указать на **/home/mike**, поэтому она здесь.

Наконец, символ доллара –приглашение для ввода. Оно значит, что мы вошли в систему как обычный пользователь; если ввести **su** для перехода в учетную запись суперпользователя (root), а затем пароль, символ доллара сменится на решетку (**#**). Попробуем ввести команду – часто это всего одно слово, например:

```
uname
```

» **Месяц назад** Мы изучили управление пакетами – RPM и Deb.

Man, помоги!

Хотите узнать больше о параметрах данной команды или программы? У большинства команд есть документация в виде man-страниц (от manual – руководство). Это не дружелюбные руководства к программам, но в них можно быстро найти значение того или иного параметра. Чтобы открыть страницу man для команды, наберите **man** и название команды; например, **man ls**. Содержимое окна просмотра можно прокручивать клавишами управления курсором; для выхода жмите Q. Для поиска определенного слова нажмите клавишу «обратный слэш» (**/**) и затем введите то, что ищете – например, для поиска слова «size» на man-странице введите **/size**.

Команда выводит название операционной системы – «Linux». Однако в закромах **uname** есть и другие возможности, и для доступа к ним служат флаги (они же – параметры, или ключи).

Флаги обычно указываются с помощью дефиса и букв или слов. Например, попробуйте выполнить команду:

```
uname -a
```

Она запустит программу **uname**, но передаст ей флаг **-a**, который означает «показать всю информацию» – и вы получите гораздо более подробный вывод. Список доступных ключей для большинства команд можно просмотреть с помощью флага **--help**, например, **uname --help**.

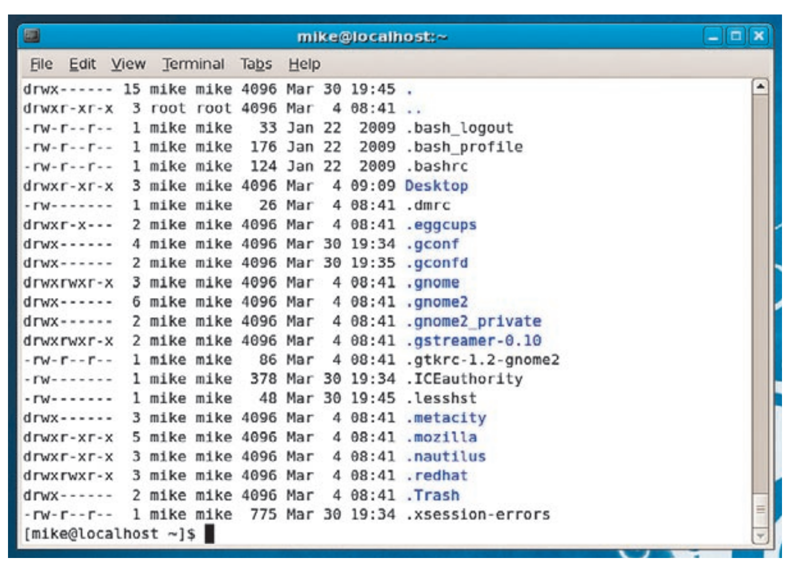
Итак, теперь вы знаете, что искать в строке приглашения, как ввести команду и как изменить ее поведение. Это все основы – теперь перейдем к управлению файлами.

Сперва введите команду **ls** – вывести список файлов. Она показывает файлы и каталоги в текущем каталоге и в зависимости от дистрибутива может выделять цветом различные элементы списка: например, подкаталоги могут быть окрашены голубым.

Команда **ls** без параметров не отображает скрытых элементов – это файлы и каталоги, чьи имена начинаются с точки. Чтобы увидеть все файлы, наберите **ls -a**. (Скрытые файлы – обычно файлы настройки, они не отображаются, чтобы не путать их с обычными файлами.) Для более подробного вывода скомандуйте **ls -l**. Учтите, что флаги можно комбинировать – например, так: **ls -l -a**, или еще короче: **ls -la**. Эта команда выводит более подробную информацию о файлах и каталогах, включая владельца, размер, дату изменения и пр.

Пока мы сидим в домашнем каталоге, но в быту администратора бывает нужно прогуляться и по другим каталогам. Прежде всего создадим новый каталог:

```
mkdir newdir
```



› Команда ls для вывода списка файлов очень гибкая, и элементы можно выводить различными способами — например, в виде подробного списка, как на рисунке.

Мы можем войти в него по команде **cd** (change directory – сменить каталог):

```
cd newdir
```

Если теперь ввести **ls**, вы увидите, что внутри каталога ничего нет. Для возврата в предыдущий каталог введите **cd ..** (cd, пробел, точка, точка).

Если в прежние времена вы пользовались DOS, вам покажутся знакомыми эти две точки – они всегда являются ссылкой на родительский каталог. Однако в отличие от DOS в команде должен быть пробел. Стоит также отметить, что в отличие от DOS все команды и имена файлов чувствительны к регистру.

Команду **cd** можно вызывать в текущем каталоге с каталогами в качестве параметров, но можно указывать и полные пути. Например, можно перейти в каталог **/usr/bin** командой

```
cd /usr/bin
```

У команды **cd** есть еще одна удобная возможность: введя **cd** без параметров, вы перейдете в домашний каталог. Это экономит время – если у вас особо развесистый логин, вам не придется набирать что-то длинное вроде **cd /home/bobthebob1234**.

Чтобы вывести полный путь до каталога, в котором вы находитесь, введите команду **pwd** (сокращение от «**print working directory** – вывести рабочий каталог»). Если вы только что сменили каталог, например, с **/usr/bin** на **/etc**, введите команду **cd —** (cd, пробел, дефис), чтобы вернуться в каталог, где вы только что были. »

$ПУТЬ к свободе

В истинных традициях Linux у этой врезки есть зависимость: основной текст статьи. Пожалуйста, сначала прочтите его, чтобы понять, что такое переменные окружения! Существует специальная переменная **$PATH**, содержащая список каталогов, из которых можно запускать программы. Введите **echo $PATH**, и увидите список этих каталогов, разделенных двоеточиями. Например, здесь есть такие каталоги как **/usr/bin**, **/usr/sbin** и т. д. (описание назначения каталогов файловой системы см. в **LXF143**). При вводе команды, например, **nano** для запуска текстового редактора *Nano*, оболочка ищет его в этих каталогах.

Однако важно заметить, что текущий каталог не является частью **$PATH**. Так сделано ради безопасности, чтобы трояны (например, зараженный исполняемый файл **ls**), проникшие в ваш домашний каталог, не запускались при каждом вызове **ls**. Если нужно запустить что-то из текущего каталога, допишите в начале команды точку и прямой слэш, например, **./myprog**. Оно, может, и нудно, но за долгие годы эта мера доказала свою важность как аспект системной безопасности Linux и Unix. Пусть вы установили что-то в **/opt**, и теперь для правильной работы путь к данному каталогу надо добавить в переменную **$PATH**. Тогда воспользуйтесь командой **export**, как описано в тексте статьи, но не сотрите текущее содержимое **$PATH** – для этого вызов должен быть таким:

```
export PATH=$PATH:/opt/newprog
```

Теперь, вызвав **echo $PATH**, вы увидите предыдущий список каталогов с добавленным в конец **/opt/newprog**.

Раздел 2: Ныряем глубже

Скорая помощь

Хотите запустить команду и закрыть сессию оболочки (например, окно терминала), когда команда закончит работу? Воспользуйтесь командой **exec** – например, **exec nano**. После выхода из текстового редактора *Nano* окно закроется.

Имена файлов и каталогов бывают длинными, особенно когда объединяются в пути, но в *Bash* есть хитроумное средство: дополнение по клавише Tab. Наберите несколько первых букв имени файла или каталога, нажмите Tab, и *Bash* попытается завершить его. Например, наберите **cd /usr/lo** и нажмите Tab – и путь должен дополниться до **/usr/local**. Если в **/usr** есть два или более каталогов, начинающихся с **lo**, *Bash* покажет все возможные варианты.

Дополнение по Tab экономит массу времени в Linux, как и история команд. С помощью стрелок вверх и вниз можно перемещаться по списку предыдущих команд (они хранятся в **.bash_history** в вашем домашнем каталоге). С помощью стрелок вправо и влево можно передвигаться по команде и редактировать ее. Введя **history**, вы увидите список последних введенных команд.

Теперь рассмотрим несколько команд для манипуляций с файлами. Чтобы скопировать файл, воспользуйтесь **cp**:

```
cp file1.txt file2.txt
```

› Не уверены, какого типа тот или иной файл? Команда file выяснит это по первым байтам файла за несколько секунд.

Можно скопировать несколько файлов в каталог командой **cp file1 file2 file3 dir**. Команда перемещения файлов работает похожим образом, и с ее помощью можно переименовать файлы: **mv oldname newname**. Чтобы удалить файл, воспользуйтесь командой **rm filename**. Но – предупреждаем: **rm** не заходит в каталоги и не удаляет то, что внутри, включая подкаталоги. Для этого укажите ключ «рекурсивно»:

```
rm -r directory
```

Эта команда удаляет каталог и все файлы и подкаталоги внутри него – ну очень мощная и разрушительная команда! (Ее альтернатива – команда **rmdir**.) Если вам попадется файл, который вы не можете идентифицировать – например, его имя не слишком наглядно или у него нет осмысленного расширения – воспользуйтесь командой **file**, которая всегда под рукой:

```
file /usr/bin/emacs
```

Эта отличная утилитка берет несколько первых байтов файла и пытается определить его тип (если это возможно). Например, обнаружив заголовок JPEG, она скажет вам, что это файл JPEG. Система, которую использует **file**, называется «волшебной» и представляет собой базу данных байтов, определяющих тип файлов. Конечно, она работает корректно не во всех случаях, и может оказаться, что простой текстовый файл определится как «База данных Microsoft FoxPro» или другая жуть, если внутри него случайно окажется определенная последовательность байтов.

Иногда бывает нужно изменить время создания файла или создать пустой файл, и здесь в дело вступает команда **touch**. Опять же, часто нужно определить местоположение файлов командной строкой, и есть два способа это сделать: **locate** и **find**. Они работают примерно одинаково, но есть одно фундаментальное различие: команда **locate foobar.txt** свяжется с заранее созданной базой данных и быстро скажет вам, где находится файл. Эта база данных обычно обновляется раз в день по расписанию программой *Cron*, поэтому информация в ней может быть неактуальной.

Для более точных результатов воспользуйтесь командой **find**, например, так:

```
find /home/mike -name hamster
```

Она выполнит доскональный поиск в каталоге **/home/mike** (и его подкаталогах) в поисках всех файлов и каталогов, содержа-

Создание и распаковка архивов

Программы, заплаты к ним и другие наборы файлов часто поставляются в виде сжатых файлов во множестве форматов. К счастью, в большинстве дистрибутивов есть необходимые утилиты для распаковки и повторного сжатия файлов, но, к сожалению, флаги у них разные. Так сложилось исторически, и поначалу это немного раздражает, но со временем вы их запомните. Вот небольшая справка:

» **.gz** Одиночный сжатый файл. Распаковывается командой **gunzip foo.gz**. Для сжатия файла воспользуйтесь командой **gzip foo**.

» **.bz2** Похож на предыдущий, но с более компактным (и более медленным) сжатием. Распаковывается командой **bunzip2**. Для сжатия файла скомандуйте **bzip2**. Этот формат с трудом работает на старых компьютерах, но на современных машинах, для распространения больших архивов с исходным кодом вроде ядра Linux, это предпочтительный вариант.

» **.tar** Архив на ленте. Лентой сейчас мало кто пользуется, но эта схема позволяет объединить несколько файлов в один (без сжатия). Распаковывается командой **tar xfv foo.tar**. Команда **tar cfv foo.tar file1 file2 dir3** создает новый архив **foo.tar** с файлами и/или каталогами внутри.

» **.tar.gz / .tar.bz2** Сочетание предыдущих форматов и самый популярный формат распространения исходного кода. Файлы объединяются в один командой *tar*, затем файл сжимается командой *gzip* или *bzip2*. Для распаковки: **tar xfv filename.tar.gz** или **tar xfv filename.tar.bz2**. Для создания архива: **tar cfvz foo.tar.gz file1 file2** (для **.tar.gz**) или **tar cfvj foo.tar.bz2 file1 file2** (для **.tar.bz2**).

» **.cpio** Относительно редкий формат, объединяющий несколько файлов в один (без сжатия). Распаковывается командой **cpio -id <filename**. Объединяет файлы команда **ls file1 file2 | cpio -ov > foo.cpio** (символ между **file2** и **cpio** – «канал», подробнее о нем в следующем месяце). Файлы CPIO могут встретиться при работе с образами initrd.

Другая полезная утилита – **dd** – копирует данные с места на место и удобна для создания образов физических носителей. Например, вставив CD или DVD и введя команду **dd if=/dev/cdrom of=myfile.iso**, вы получите ISO-образ (которым можно с кем-то поделиться или записать его на другой диск).

» **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.

щих «hamster» в своем имени. Но что если вы намереваетесь произвести поиск в текущем каталоге? Вряд ли вам захочется набирать его полный путь. Выход есть: помните, ранее мы упоминали, что **..** (две точки) – родительский каталог по отношению к текущему. А вот **.** (одна точка) – это текущий каталог. Поэтому предыдущую команду можно переписать так, учитывая, что вы уже находитесь в каталоге **/home/mike**:

```
find . -name hamster
```

В команде **find** также можно оговорить размер файла: **find . -size +100k** ищет все файлы размером больше 100 килобайт в текущем каталоге (используйте **M** для МБ и **G** для ГБ). Другая альтернатива – поиск по типу: **find . -type f** найдет только файлы, а **-type d** – только каталоги. Опции -name, -size и -type можно объединять для конкретизации условий поиска.

В *Bash* есть исчерпывающие возможности для указания маски нескольких файлов без необходимости перечислять их имена. Звездочка (*****), например, означает «любая комбинация букв, цифр и других символов». Попробуйте выполнить команду

```
ls *.jpg
```

Она выведет список всех файлов с расширением **.jpg**, будь это **bunnyrabbit.jpg**, **4357634.jpg** или что-то еще. Ею удобно пользоваться для перемещения и удаления файлов: если у вас есть каталог с изображениями и вы хотите избавиться от тех, что с расширением **.bmp**, можно выполнить команду **rm *.bmp**. Если нужна маска только для одного символа, воспользуйтесь знаком вопроса:

```
mv picture?.jpg mypics
```

Эта команда переместит **picture1.jpg**, **pictureA.jpg** и т.д. в каталог **mypics**.

Скорая помощь

Если вы ввели команду, которая, кажется, закончит работу через несколько часов, и хотите остановить ее, нажмите Ctrl+C. Помните, что эта комбинация клавиш прерывает выполнение команды незамедлительно, поэтому есть вероятность, что она не очистит файлы, с которыми работала!

Раздел 3: Разберемся в окружении

Команды, хотя обычно мы набираем их одну за другой, зависят от среды, в которой работают. Существуют переменные окружения, которые хранят информацию об опциях и настройках, и программы могут брать эту информацию, чтобы понять, как им работать. Переменные окружения обычно задаются заглавными буквами и начинаются со знака доллара. Например, попробуйте выполнить

```
echo $BASH_VERSION
```

Здесь **echo** – команда, которая просто выводит текст, в данном случае значение переменной **$BASH_VERSION**, на экран. Вы увидите число вроде 3.2.25. Программы могут проверять значение переменной в собственных целях – например, чтобы узнать, запущена ли версия 3.0 или новее и, следовательно, есть ли в ней определенные функции. Чтобы увидеть полный список используемых переменных окружения с их значениями, введите **env**.

```
mike@localhost:~
File Edit View Terminal Tabs Help
MAIL=/var/spool/mail/mike
PATH=/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/home/mike/bi
n
DESKTOP_SESSION=default
GDM_XSERVER_LOCATION=local
INPUTRC=/etc/inputrc
PWD=/home/mike
XMODIFIERS=@im=none
LANG=en_US.UTF-8
GDMSESSION=default
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
SHLVL=2
HOME=/home/mike
GNOME_DESKTOP_SESSION_ID=Default
LOGNAME=mike
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-f5xEc8fxce,guid=0588delbde46afe
c21af12084d94351b
LESSOPEN=|/usr/bin/lesspipe.sh %s
DISPLAY=:0.0
G_BROKEN_FILENAMES=1
COLORTERM=gnome-terminal
XAUTHORITY=/tmp/.gdmFFAGTV
_=/usr/bin/env
[mike@localhost ~]$
```

› Переменные окружения изменяют способ работы программ — получите их полный список командой env.

Можно создать и свою переменную среды:

```
export FOO="bar"
echo $FOO
```

(Обратите внимание, что в первой команде нет знака доллара.) Новая переменная **$FOO** будет существовать, пока открыт сеанс терминала; когда вы закроете его, набрав **exit** или нажав Ctrl+D, переменная исчезнет. Чтобы сохранить переменную, добавьте ее в файл **.bashrc** в вашем домашнем каталоге. В этом файле содержатся определения переменных и другие настройки, которые считываются в начале нового сеанса. Сохраните изменения, перезапустите терминал, и они войдут в силу.

В *Bash*, кроме переменных окружения, в котором выполняются программы, есть и другие переменные – его собственные. Введите команду **set**, и вы увидите их полный список. Удалить переменную, будь то переменная *Bash* или переменная окружения, можно следующимобразом:

```
unset FOO
```

Эти возможности, в сочетании с дополнением по Tab, символами маски и историей команд, делают командную строку Linux необычайно эффективной, особенно по сравнению с неуклюжей командной строкой DOS из далекого прошлого. Чем ближе вы будете знакомится с командной строкой, тем больше будет соблазн покинуть файловый менеджер.

Главное, ситуация всегда у вас под контролем. При работе в командной строке нельзя случайно выбрать неверную опцию (не считая опечаток): вы делаете именно то, что хотите. Впрочем, это только часть истории – в следующий раз мы научимся перенаправлять вывод с одного терминала на другой или в файл для последующего просмотра. Не пропустите! LXF

Скорая помощь

Случайно заполнили терминал непонятными символами? Для очистки экрана введите команду **clear** (или нажмите Ctrl+L). Если она не работает и странные символы продолжают появляться из-за того, что на терминал вываливаются случайные двоичные данные, попробуйте **reset**.

Проверь себя!

Думаете, что усвоили темы и команды, которые мы сегодня обсудили? Хотите понять, готовы ли вы пользоваться ими на экзамене LPI или в реальной жизни? Посмотрите, сможете ли ответить на эти вопросы, и переверните журнал, чтобы прочесть ответы на них, внизу:

1. Что означает тильда (~) в строке приглашения?
2. Как вывести подробный список всех файлов в текущем каталоге?
3. Какая команда поиска файлов использует предварительно созданную базу данных?
4. Как установить значение переменной окружения **$WM** в **icewm**?
5. Как добавить **/opt/kde/bin** к вашему **$PATH**?
6. Как создать архив **.tar.bz2** из каталога **myfiles**?
7. Вы хотите запустить версию *Nano* из текущего каталога, а не из каталога, указанного в **$PATH**. Как это сделать?

1 — Домашний каталог. 2 — ls -la. 3 — locate. 4 — export WM="icewm". 5 — export PATH=$PATH:/opt/kde/bin. 6 — tar cjvf archive.tar.bz2 myfiles. 7 — ./nano.

» Через месяц Сведем более близкое знакомство с командной строкой.

Команды: GNU/

Тихон Тарнавский описывает необычные приемы и навыки работы с командной строкой – сравнив их с выкладыванием мозаики.

Наш эксперт

Тихон Тарнавский
Работает в Linux-консоли больше восьми лет и точно знает, как сделать эту работу удобной.

Идея этой статьи зрела долго – почти пять лет. Ведь написано о командной строке за годы существования Unix не просто много, а невообразимо много; и материалы разнятся как по объему, так и по необходимому для прочтения уровню подготовки и стилю изложения. Исходя из этого, многим может быть непонятно: зачем же браться за ту же тему сызнова? Потому для начала – небольшое объяснение.

Что это, зачем и для кого?

Уникальность предлагаемых заметок в том, что в них я не собираюсь учить вас, уважаемый читатель, работе в командной строке. Вместо того, чтобы сосредоточиться на обучении, я предлагаю вам попытаться понять, чем же так ценна та самая командная строка многим опытным линуксоидам. Да, именно так: понять самим, но с моей помощью. Я не буду вам этого объяснять; объяснить своими словами эти вещи тоже уже пытались многие, но каждый раз на этом пути встает известная проблема понимания: человек, способный уже объяснить любую познаваемую своим опытом истину, говорит на другом языке, чем те, к кому он обращается – именно потому, что они сами этого опыта еще не имеют.

Не будет здесь и методов «от простого к сложному» – вместо этого вам предлагается понять, что ничего сложного здесь нет. Одним словом, попробую отбросить все каноны изложения «основ» и «принципов», а взамен воспроизведу сам процесс. Надеюсь, из этого вы сможете понять, что чувствует человек, имеющий в своих руках столь мощный, гибкий и одновременно простой инструмент. Именно на этом – на ощущении процесса – и предлагаю вам сосредоточиться во время чтения.

> «Попробую отбросить все каноны изложения основ и принципов.»

Материал будет состоять из практических примеров: иногда совсем небольших, иногда – достаточно объемных. Каждый пример будет снабжен объяснением, но вся суть материала будет в построении этих объяснений. Обычно такие вещи разъясняются «от конструкции»: уже готовый набор команд разбирается последовательно, с пояснением, что же каждый элемент делает. Я же предлагаю пойти противоположным путем: «от задачи». В каждой заметке мы будем ставить перед собой некую задачу, которую человек может захотеть решить, работая в командной строке, и подробно рассматривать ход мыслей этого человека и их практическую реализацию до конечного результата. Именно такой подход, на мой взгляд, и даст почувствовать всю ту красоту средств командной строки, которую, поняв однажды, ни на что уже не захочется променять.

```
$ du | sort -n | tail
66052   ./files/deb/repack/emacs-nox/usr
75652   ./files/deb/repack/emacs-nox
78064   ./pocketbook/themes
105584  ./pocketbook
115952  ./info
162024  ./files/deb/repack
200888  ./files/deb
201356  ./files
692060  ./lit
1263176 .
$ █
```

› **Рис. 1. Находим самые объёмные каталоги.**

Предназначаются заметки для самого широкого круга читателей, которым будет интересно расширить свои познания в этом вопросе – да и в Linux вообще, поскольку команды, программы и приемы, применяемые при работе в командной строке, составляют основу системы. Если вы только начинаете свое знакомство с Linux, для вас в небольших врезках предусмотрены краткие пояснения даже по самым простым вещам. Если имеете определенный (может быть, даже очень большой) опыт, можете такие объяснения (или некоторые из них) пропускать – целостность остального текста от этого нисколько не проиграет.

Корни

Для успешного роста нужна связь с землей, и некоторую базу в начале придется изложить, но она совсем невелика, потому, надеюсь, я вас не утомлю. Базовых идей, которые у тех, кто не понял их сразу, вырабатываются опытом, и которые именно для облегчения практики рекомендуется знать с самого начала, всего две:

1 **«Большой груз легче перенести по частям»** Задач, которые человек мог бы захотеть решить с помощью компьютера, великое множество. Поэтому делать по отдельной программе на каждую задачу нерационально – тогда программистом пришлось бы быть каждому. Вместо этого предлагается разбить большую задачу на маленькие кусочки и каждый из таких кусочков решить уже готовым специализированным инструментом. Инструмент этот, ввиду узости применения, достаточно мал и прост для использования его без специальной подготовки. Для объединения этих маленьких деталей в большие конструкции предоставлен достаточно широкий спектр связей. Практика такого разделения задачи будет подробно рассмотрена в каждом конкретном примере.

2 **«Не знаешь (не помнишь) — читай»** Все команды и все их параметры в голове удержать не под силу даже тому, кто использует их ежеминутно. Потому не гнушайтесь чтением man-страниц, благо написаны они обычно очень коротко и просто, и на нынешний момент очень многие из них есть на русском языке. Для облегчения усвоения информации и последующего ее поиска (когда забудется) рекомендую (по крайней мере, на первых порах) составлять глоссарий с наиболее часто используемыми командами.

Устраиваемся поудобнее

В Linux много различных командных оболочек; здесь я буду говорить исключительно о *bash* (Bourne Again Shell), по двум причинам: с одной стороны, она самая распространенная (и используется по умолчанию почти во всех дистрибутивах), с другой – одна

Linux и смекалка

из двух самых мощных и гибко настраивамых, а следовательно, и самых удобных (наравне с *zsh*).

Для начала и предлагаю свою оболочку настроить. Две основные вещи, которые лучше сразу сделать более функциональными, это история команд и автодополнение. Для первого добавим в файл **.inputrc** в домашнем каталоге такие строки (если такого файла нет, создайте его):

```
"\e[A": history-search-backward
"\e[B": history-search-forward
```

После этого, набрав в командной строке один или несколько символов, вы сможете пробегать стрелками вверх-вниз не всю историю, а только те команды, которые начинаются с этих символов. Поиск по всей истории тоже остается доступным: по нажатию вверх или вниз в пустой командной строке.

В **.bashrc** рекомендую добавить такую строку:

```
HISTCONTROL=ignorespace:erasedups
```

Первый из этих параметров запрещает сохранение в истории строк, начинающихся с пробела: теперь для того, чтобы не засорять историю лишними командами, достаточно ставить перед ними пробел. Второй параметр заставляет в момент добавления новой команды к истории удалять оттуда все ее более ранние вызовы – опять же для лаконичности истории.

На других удобных настройках (касающихся не только истории) я не буду подробно останавливаться; ищите их вместе с уже описанными в файлах **.bashrc** и **.inputrc** на диске, приложенном к журналу. Эти файлы нужно положить в домашний каталог, после чего все заданные в них настройки будут доступны для всех новых терминальных окон. А если какие-то из этих настроек окажутся актуальны по ходу повествования, я буду в двух словах пояснять, что они означают.

Для умного автодополнения достаточно установить пакет *bash-completion* (если такой отдельный пакет есть в вашем дистрибутиве, а не включен в состав пакета *bash*) и активировать запуск скрипта из этого пакета в том же файле **.bashrc**.

Рекомендую также обратить внимание на приведенные во врезке сочетания клавиш *bash*. Например, узнав о наличии специального сочетания, меняющего местами два соседних символа, я не очень понял, зачем оно нужно; но потом обнаружил, что при достаточно быстром наборе перепутанные местами символы – одна из наиболее распространенных опечаток.

Принципы

Вы, думаю, знаете, что традиционный инструментарий Unix (а соответственно, и GNU/Linux) состоит из множества небольших узкоспециализированных утилит. Главное – не подумать, что для работы в командной строке нужно изучать все эти утилиты. Это наименее эффективный подход, хотя, как ни странно, и наиболее распространенный. Напротив, не сосредотачивайтесь на отдельных командах. Поначалу лучше их вообще не стараться специально запоминать, а записывать в глоссарий, о котором я уже писал выше. Смотрите на них как на конструктор или мозаику. Вы в детстве играли с конструктором или мозаикой? Представьте,

```
$ find -type f -print0 | xargs -0 du | sort -n | tail
2744    pocketbook/src/fbreader-src-260609.rar
2824    pocketbook/htdocs/files/demo.zip
3284    files/deb/mueller7-dict_2002.02.27-7_all.deb
3396    files/deb/mueller7accent-dict_2002.02.27-5_all.deb
4228    files/deb/pidgin-data_2.5.6-1_all.deb
17472   info/backup.tar.gz
17524   doc/79.pdf
31664   files/deb/emacs-nox_22.1-1_armel.deb
31904   files/deb/emacs_22.1-1_armel.deb
61820   old/pb360.tar
$
```

› Рис. 2. Находим самые увесистые файлы.

что во время игры вы бы старательно изучали каждую деталь или каждую фишку. Было бы это интересно или хотя бы результативно? Помогло бы вам такое изучение сложить узор или собрать машину? Утилиты Linux, конечно, немного сложнее по своему устройству, чем мозаичные фишки или детали конструктора – но предназначение у них в большинстве случаев в точности то же: стать частями общей картины или механизма. Поэтому главное, чему и нужно уделить внимание – это связи, взаимодействие между разными командами.

Основных принципов Unix и в целом немного, но нам, по крайней мере на первых порах, будут интересны всего два из них, причем сильно взаимосвязанных. Первый из этой пары: «все есть текст». Именно так: не только информация, которую мы получаем от большинства команд, но и их управляющие параметры, и «данные», которые они обрабатывают – все это обычный текст. И не только это, но и информация о процессах (работающих программах), файлах на вашем диске и правах доступа к ним, «железной» начинке вашего компьютера и подключенных к нему внешних устройствах доступна именно в текстовом виде. Все это сделано именно так сознательно, для удобства обычного человека, делающего свою работу с помощью компьютера.

Второй, смежный, принцип: «если вы не хотите читать вывод программы, заставьте это делать другую программу». Собственно, »

История GNU

В 70-е годы многие компьютерные программы были свободными, что и поспособствовало бурному развитию отрасли. Но к началу 80-х ситуация изменилась: все больше компаний стали диктовать пользователям своих программ, что они могут с этими программами делать. Тогда сотрудник Массачусетского технологического института (MIT, Massachusets Institute of Technology) Ричард Столлмен решил разработать полностью свободную операционную систему, технически совместимую с Unix. Так возник проект GNU (GNU's not Unix, GNU –это не Unix), в рамках которого разработаны все утилиты командной строки, которые мы будем использовать, включая командную оболочку *bash* (а также графическая библиотека *GTK*, основанное на ней окружение Gnome, текстовые редакторы *Emacs* и *TeXmacs* и многое другое). Первоначально о проекте было объявлено в сентябре 1983 года.

Подробнее см. на сайте проекта: www.gnu.org/gnu/gnu-history.ru.html.

Другие интересные ссылки:

» http://www.gnu.org/philosophy/free-sw.ru.html

» http://www.gnu.org/philosophy/philosophy.ru.html

› **Рис. 3. Процессы, активно использующие процессор.**

```
$ ps aux | sort -nk 3 | tail -3
user      1135  3.8 13.9 45864 17696 ?        S<s  08:07  25:28 /usr/bin/hildon-input-method
user      1417  4.5  9.7 38748 12308 ?        Ss   08:08  30:24 /usr/bin/osso-xterm
user      3115 99.9  6.0 27572 7672 ?         Rs   19:17   0:01 pidgin
$
```

этот принцип и заключается в себе описанную выше аналогию с конструктором или мозаикой: это те самые «винтики», которыми можно соединить отдельные «детали»; или те «ячейки», в которые вставляются отдельные «фишки».

Но хватит теории: посмотрим, как все это выглядит в реальной жизни. Запускайте терминал – и приступаем.

Работа с файлами: оценка объемов

Давайте начнем с достаточно простой задачи: узнаем, сколько места занимают на диске наши файлы. Понадобиться это может много для чего. Например, вы срочно захотели скачать приложение, и требуется либо найти что-нибудь ненужное и удалить, либо найти что-нибудь нужное, но «не сейчас, а в принципе», и сбросить на какой-нибудь внешний носитель. Естественно будет поиск ненужного начать с самых больших файлов или каталогов. Чтобы узнать, что сколько весит, воспользуемся командой **du** (du – сокращение от “disk usage”, «использование диска»). Начать можете с вызова ее без параметров; и посмотрите, что получится:

```
$ du
16      ./.mc
24      ./.ssh
404     ./bin
4      ./.osso-abook/extensions
16      ./.osso-abook/db
24      ./.osso-abook
8      ./.osso/mis
8      ./.osso/menus
12      ./.osso/hildon-input-method
36      ./.osso/hildon-desktop
[...]
51716   ./files/deb
52184   ./files
965120  .
```

Как видите, вывод не слишком удобен для дальнейшего применения: **du** прошлась вглубь по всем каталогам и «взвесила» каждый из них. Выискивать среди них самые большие вручную – занятие не легче, чем найти нужные (или ненужные) файлы безо всяких **du**. Вот здесь и пора применить озвученный принцип: заставить работать вместо нас другую программу. Хорошо было бы отсортировать этот вывод, чтобы самые большие числа шли в конце и мы их сразу легко увидели. Для сортировки существует специальная команда, и называется она весьма очевидно – **sort**. Так что даже специально искать не нужно. Но по умолчанию она сортирует текстовые строки в алфавитном порядке, а нам нужно отсортировать числа. Поэтому для начала откроем **man sort** и найдем нужную опцию: **-n** – сортировать в соответствии с числовыми значениями строк (n от слова numeric; numeric sort – числовая сортировка). Осталось передать полученный выше текст нужной команде для сортировки. Для этого в shell есть специальный «винтик» – конвейер. Обозначается он вертикальной чертой, которая ставится между командами и передает вывод первой на вход второй. Можно было бы сразу попробовать такой конвейер:

```
du | sort -n
```

но по выводу команды **du** вы, думаю, увидели, что результат очень велик. Если он уже будет отсортирован, то хорошо бы оставить только пару десятков последних строк. Как это сделать? Вся мощь конвейеров в том, что мы можем объединять не только две команды, а сколько угодно: ведь вторая, получив от первой некий текст, сделает из него опять-таки текст. А любой другой команде не важно, как этот текст создан – одной командой или двумя; или он вообще лежит в файле или набирается с клавиатуры. Давайте попытаемся найти команду для вывода последних строк. Тут нам снова пригодится команда **man**: у нее есть специальный ключ для поиска ключевых слов по описаниям man-страниц – **-k** (keyword – ключевое слово). Узнать о нем можно из руководства к самой этой команде – **man man**. Используем этот ключ со словом last (последние):

```
$ man -k last
tail (1)  - output the last part of files
```

Добавим команду **tail** в конец нашего конвейера (см. рис. 1).

Как видите, по умолчанию мы получили десять строк. Если этого мало, количество строк можно изменить с помощью опции к команде **tail**, например **tail -n 20**, или сокращенно **tail -20**.

Уже достаточно удобно, если нужно искать большие каталоги. А как быть с обычными файлами? У команды **du** есть ключ **-a** (all – все, т. е. не только каталоги, но и обычные файлы), но здесь он мало поможет: какие-то каталоги скорее всего будут весить больше и вытеснят файлы из последних строк списка. Например, у меня от добавления этой опции в вышеприведенном конвейере вывод совсем не меняется. Команда **du** предназначена именно для работы с каталогами, и подсчет «тяжести» обычных файлов – лишь приятный довесок, не более. Как быть? Очевидно, нужно начать с критичной для нас части этой задачи. То есть сначала найти обычные файлы, а потом уже готовый список передать команде **du**, как только что мы ее вывод передавали командам **sort** и **tail**.

Значит, нам нужен поиск по типу файла (здесь важно не путать тип файла с «расширением» – понятием, более свойственным семейству Windows, а не Unix; поскольку в Unix и каталоги, и файлы устройств, и много других сущностей являются файлами, то именно каждое из этих понятий и есть отдельный тип файла). Имя команды поиска столь же очевидное, как и команды сортировки: **find** (находить). Имя опции для поиска по типу файла не менее очевидно: **-type** (тип). А значение аргумента, который эта опция принимает, состоит из одной буквы, и для обычного файла эта буква также совершенно ожидаема: **f** (file).

```
find -type f
```

man-страницы в Сети по-русски

Если вам сложно воспринимать информацию на английском языке, удобно будет использовать русскоязычные man-страницы. Вполне возможно, что они доступны в вашем дистрибутиве в пакете, названном **manpages-ru** или как-то похоже. Но не исключено, что пакет этот недостаточно полон или вообще отсутствует. В таком случае попробуйте найти нужную man-страницу на этом сайте: http://opennet.ru/man.shtml.

```
$ ps aux | sort -nk 6 | tail -5
user      1417  4.5  9.7 38748 12308 ?      Ss   08:08  31:32 /usr/bin/osso-xterm
user      1177  1.7 11.3 54288 14432 ?      S<s  08:07  11:51 /usr/bin/hildon-desktop
user      3198 17.2 14.1 43096 17956 ?      Ss   19:36   0:03 /usr/bin/midori
user      1135  3.8 14.3 46384 18248 ?      S<s  08:07  26:35 /usr/bin/hildon-input-method
user      3196 25.1 15.2 43332 19316 ?      Ss   19:36   0:07 pidgin
$ 
```

› Рис. 4. Процессы, занимающие много памяти.

Теперь нужно передать результаты поиска команде **du**. Сделать это напрямую по конвейеру на этот раз не получится, т. к. **du** принимает список файлов из командной строки, а не со стандартного ввода. Но именно на этот случай в комплекте с **find** (в том же самом пакете *findutils*) есть не менее богатая возможностями программа **xargs**, которой в простейшем варианте нужно передать только имя команды (и опции, если они нужны), и она сама подставит свой стандартный ввод в командную строку:

```
find -type f | xargs du
```

Теперь продолжим полученную цепочку тем же способом, что и в первый раз:

```
find -type f | xargs du | sort -n | tail
```

Все уже будет как задумано – но только если у вас нет файлов с пробелами в именах. Иначе каждый такой пробел будет воспринят командой **xargs** как разделитель аргументов командной строки, и вместо размера файла вы получите столько сообщений о ненайденных файлах, на сколько «слов» пробелами разбивается имя. Но не переживайте, это тоже предусмотрено авторами *findutils*: велите команде **find** использовать в качестве разделителя имен «нулевой» символ, а команде **xargs** – что вывод нужно делить на отдельные имена только «нулевыми» символами (см. рис. 2). Теперь найти большие файлы или каталоги легко.

Если вы предполагаете, что какие-то команды могут быть нужны вам достаточно часто, то было бы удобнее их сократить при вводе. Например, для только что созданных нами конвейеров можете добавить вваш **.bashrc** такие строки:

```
alias bigdirs='du | sort -n | tail -20'
alias bigfiles='find -type f -print0 | xargs -0 du | sort -n | tail -20'
```

Хотя, чтобы чувствовать себя свободнее в добавлении таких сокращений и в то же время не засорять стартовый файл *bash*, в который может быть нужно добавить другие удобные настройки, предлагаю вынести подобные вещи в отдельный файл (например, **.bash_aliases**), а из **.bashrc** его вызывать. Заодно можно будет легко применять новые добавления в этом файле:

```
. ~/.bash_aliases
```

Выполнять в уже работающей сессии весь **.bashrc** я не очень рекомендую, т. к. в нем могут быть и такие вещи, повторный запуск которых в том же сеансе приведет к непредсказуемым результатам.

> «Если команды нужны вам достаточно часто, можно их сократить.»

Работа с системой: оценка ресурсов

Представим другой критический случай: система вдруг начала «тормозить». Знакомо? Происходить это может по двум причинам: либо какой-то процесс (работающая программа) слишком сильно нагрузил процессор, либо стал использовать очень много памяти. Обе эти ситуации (да и многое другое, что касается процессов) можно отследить с помощью команды **ps** (processes snapshot, снимок процессов). Здесь нам достаточно знать, что наиболее полную информацию о процессах можно получить командой **ps aux**. Дальше, думаю, понятно: мы снова должны отсортировать и взять последние строки. Только как сортировать? Давайте посмотрим на заголовки столбцов, которые выводятся в первойстроке:

```
$ ps aux | head -1
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
```

То есть, чтобы отыскать самых прожорливых поглотителей процессора, нам необходимо отсортировать список процессов по третьему столбцу. Понятно, что тут должна помочь команда **sort**. Применим ключ **-k** (keyfield, ключевое поле) и числовую сортировку:

```
ps aux | sort -nk 3
```

Скорее всего нам будет достаточно двух-трех самых жадных процессов. Добавляем **tail -3** и безошибочно находим виновника – *pidgin* (см. рис. 3).

Самый общеизвестный и безжалостный способ решения такой проблемы – избавиться от найденного виновника раз и навсегда командой **killall pidgin** (или даже **killall -KILL pidgin**, если не послушается). Но если мы в курсе, что программа занята чем-то нужным, а нам просто понадобилось что-то заархивировать, можно мешающий процесс приостановить, а затем продолжить:

```
$ killall -STOP pidgin
$ tar cf archive.tar.lzma --lzma dir/
$ killall -CONT pidgin
```

Теперь понятно, что для выяснения ситуации с памятью можно поступить аналогично (только «хвост» оставим чуть подлиннее). Но возьмем не четвертое поле, а шестое: в нем отображается тот же объем, но не в процентах, а в килобайтах, т. е. немного точнее (см. рис. 4). LXF

Обозначения сочетаний клавиш

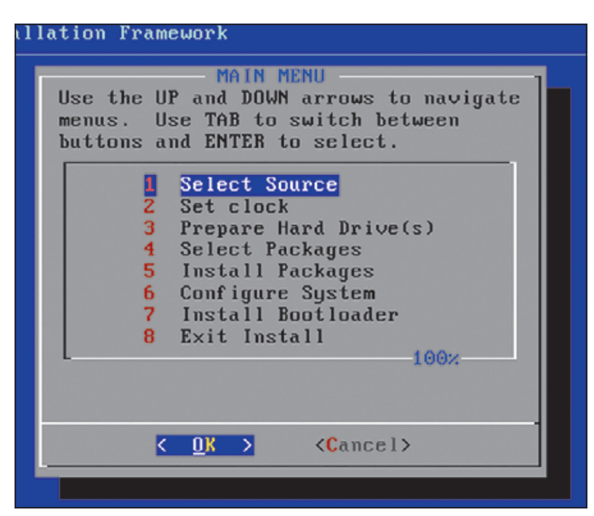
В *bash* приняты обозначения сочетаний клавиш, которые некоторым могут быть непривычны, потому приведу их расшифровку. Сокращение **C-a** означает **Ctrl+a**. Сокращение **M-a** означает **Alt+a** (клавиша, которую сейчас обозначают как Alt, раньше называлась Meta; имя Alt использовалось для другой клавиши, которой на большинстве современных клавиатур нет) либо последовательное нажатие **Esc**, **a**.

Сочетания клавиш	Пояснение
Перемещение курсора	
M-f	На слово вперед (forward)
M-b	На слово назад (backward)
Редактирование строки	
C-w, M-BackSpace	Удалить до границы слова назад
M-d	Удалить до границы слова вперед
C-k	Удалить до конца строки
C-u	Удалить до начала строки
C-t	Поменять местами (transpose) два соседних символа
M-t	Поменять местами два соседних слова
История	
C-r	Поиск от конца истории к началу (reverse search – обратный поиск)
M-.	Подставить на место курсора последний аргумент предыдущей команды; следующие нажатия «проматывают» последние аргументы назад по истории

Arch Linux:

Устав от однообразия дистрибутивов и тоскуя по новизне, **Шашанк Шарма** набрел на Arch и опять почувствовал вкус к Linux.





› Внимательно читайте инструкции в верхней части каждой страницы, чтобы знать, какую клавишу когда нажимать.



Наш эксперт

Шашанк Шарма более четырех лет пишет о свободном ПО для различных изданий, включая Linux.com. Он соавтор книги «Beginning Fedora».

В последние годы разработчики большинства дистрибутивов Linux делали упор исключительно на работоспособность, и это, несомненно, вдохновило многих шагнуть в мир Linux. Arch Linux – не для этих новичков. В отличие от большинства дистрибутивов, Arch не навязывает своим пользователям никаких пакетов и приложений.

Все компоненты системы (кроме базовой версии ядра, которая тоже настраивается) – в руках пользователя. Это означает, что вам не видать ни рабочего стола, ни оконного менеджера, не говоря уже о браузере и текстовом редакторе, если вы не установите все это сами. После установки по умолчанию будет создан только пользователь root – всех остальных придется добавлять вручную командой **useradd**. Это неотъемлемая часть философии Arch: хранить систему простой, давая пользователям возможность самим выбрать компоненты, составляющие их систему. Этот пользователе-центристский дизайн требует подхода «сделай сам».

> «Arch не навязывает своим пользователям пакеты и приложения.»

Рассмотрим ваши варианты

В дополнение к базовым пакетам вы получаете огромное количество конфигурационных скриптов, которые нужно изменить, чтобы различные программы и устройства заработали. Например, чтобы заработали сетевые карты, нужно, кроме установки должных драйверов и приложений, изменить несколько файлов настройки. В результате Arch часто называют дистрибутивом для тех, кто не боится командной строки.

Другая отличительная черта, противопоставляющая Arch другим дистрибутивам – выпуск релизов. У большинства дистрибутивов новые релизы и установочные диски выходят через определенные промежутки времени, но в Arch обновления ПО выходят регулярно. Чтобы воспользоваться последним релизом, не нужно переустанавливать систему, в отличие от большинства других дистрибутивов: нужно просто ее обновить. Это позволяет Arch согласовываться с графиком релизов ядра Linux – так обеспечивается лучшая поддержка устройств. Быстрое обновление после установки текущего релиза Arch от мая 2010 года приведет вашу систему в актуальное состояние.

Pacman – менеджер пакетов Arch и, пожалуй, один из секретов его успеха. Благодаря *Pacman* систему можно установить практически мгновенно, несмотря на необходимость устанавливать все утилиты и приложения вручную, но только если у вас хорошее соединение с Интернетом.

Способный разрешать зависимости *Pacman* умеет опрашивать репозитории и обновлять, устанавливать или удалять любые пакеты. В отличие от других подобных утилит он работает со сжатыми двоичными пакетами в формате **tar.gz** или **tar.xz**. В файлах **/etc/pacman.conf** описываются различные настройки *Pacman*, а также используемые репозитории. Основных репозиториев три: Core, Extra и Arch User Repository (AUR). Прежде чем нырять в мир Arch, советуем узнать о *Pacman* как можно больше. Хорошая стартовая точка – вики по Arch (см. врезку о документации по Arch), но также стоит просмотреть соответствующую man-страницу.

По умолчанию все зеркала в файле **/etc/pacman.d/mirrorlist** отключены. Удалив символ **#** перед зеркалом, вы его включите. Что-

Сделай сам



бы понять, каким зеркалом лучше пользоваться, потребуется время, так как в данный момент времени зеркало может не работать. Включив зеркала, можно обновлять систему с помощью *Pacman*.

Команда **pacman -Sy** обновит список пакетов, после чего можно установить пакеты командой **pacman -S package-name/**. Если вы хотите удалить пакет, укажите ключ **-R**. Команда **pacman -Syu** обновит все пакеты. Для поиска пакета воспользуйтесь командой **pacman -Ss package-name**, а с ключом **-Q** можно спросить об установленных пакетах. Позже, после установки дистрибутива, мы увидим *Pacman* в действии.

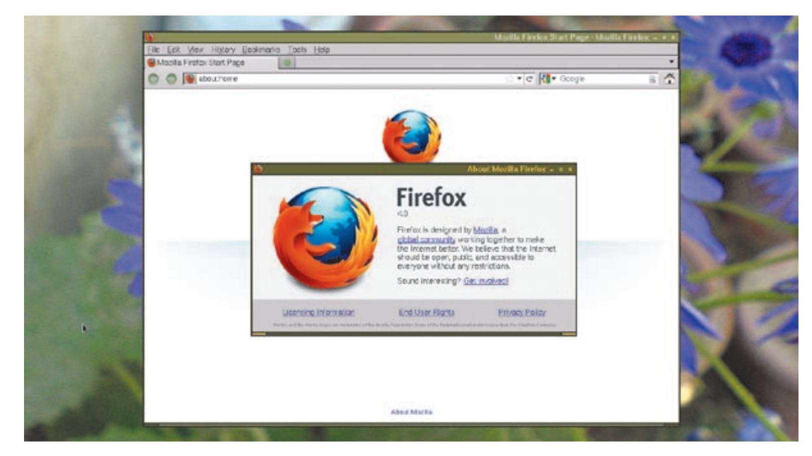
Установщик Arch

Проще всего установить Arch из ISO-образа. Его можно установить и из рабочей системы Linux, но это гораздо сложнее; если хотите пойти этим путем, обратитесь к вики по Arch. Даже в случае с ISO-образами возможны два варианта: базовый образ и сетевой образ. Сетевой образ не содержит пакетов – он полагается на ваше интернет-соединение, через которое будут загружены последние версии пакетов и сформирована самая свежая система. Мы будем устанавливать Arch из образа ядра. Если у вас нет неприлично быстрого интернет-соединения, советуем устанавливать систему из базового образа. Как во всех дистрибутивах, образ нужно прожечь на диск и загрузиться с него.

Установщик Arch, *Arch Installation Framework* (*AIF*), написан на *Bash* и использует *Dialog* для вывода текстовых окон в процессе установки. Для перемещения по кнопкам пользуйтесь клавишей Tab, по пунктам меню – стрелками вверх и вниз, а для выбора пункта меню жмите Enter. После загрузки с прожженного диска зайдите в систему как пользователь root. Пароль вам не понадобится. Запустите установку командой: **arch/setup**.

Основных задач семь. Во-первых, укажите источник установки. Если у вас быстрое интернет-соединение, выберите "Net [Сеть]". В противном случае продолжите установку с диска, выбрав вариант "CD-ROM". Вторая задача, установка времени [Set Clock], вполне стандартна: нужно выбрать свой часовой пояс и установить время. Сетевая карта еще не настроена, и воспользоваться протоколом сетевого времени (NTP) нельзя – вы должны будете сделать это вручную. По завершении каждого этапа вы будете возвращаться в главное меню установки, где будет выбран следующий шаг. Для продолжения установки нажимайте Enter.

Третий этап – подготовка жесткого диска для установки. Варианты выбора сегодня стандартны для большинства дистрибутивов – использовать диск целиком, определить разделы вручную и т. д. В ручном разбиении диска на разделы поможет утилита *Cfdisk*. Если вы не пользовались ею раньше, потребуется некоторое время на освоение. Создайте разделы, вернитесь на экран «Подготовка жесткого диска» (Prepare Hard Drive) и вручную настройте блочные устройства, точки монтирования и т. д., что влечет за собой определение файловой системы для каждого



› **В репозиториях дистрибутива находятся самые свежие версии программ — для установки *Firefox 4* в Arch Linux достаточно набрать pacman -S firefox.**

из вновь созданных разделов, таких как **/** и **swap**. Если вы отдадите в распоряжение Arch весь диск, он создаст четыре раздела: **boot**, **swap**, **/** и **/home**, давая вам возможность задать размер каждого из разделов, кроме **/home**, который получит остаток, когда все остальные размеры будут заданы. Затем выберите файловую систему для разделов **/** и **/home**.

Теперь мы подошли к четвертому этапу – выбора пакетов для установки. По умолчанию выбран только пакет **Base**. Выберите еще **Base-devel**, затем нажмите Enter. Появится список пакетов – выберите те, которые хотите установить. В **Base-devel** вы найдете в списке такие пакеты как **make** и **gcc**, и они будут уже выбраны. Все заранее выделенные пакеты в каждой из этих категорий советуем оставить. Затем пакеты установятся, что ознаменует окончание пятого этапа установки.

Как мы отмечали, вам придется основательно углубиться в настройки, особенно в свежеустановленной системе Arch, поэтому »



Релиз Arch по сути представляет собой снимок текущих пакетов в репозитории Core.

Репозитории Arch

Репозитории Arch можно разделить на две категории – официальные репозитории и репозитории пользователей Arch:

» **Core** Содержит все утилиты, необходимые для получения полностью рабочей основной системы.

» **Extra** Содержит пакеты, которые не попадают в Core, такие как оконные менеджеры, рабочие столы, медиа-проигрыватели и т. д.

» **Testing** Здесь находятся последние версии пакетов – претендентов на место в Core или Extra.

» **Community** Содержит пакеты, собранные и одобренные сообществом.

» **Community-testing** Напоминает Testing, отличие лишь в том, что его пакеты являются кандидатами на попадание в Community.

» **Multilib** Репозиторий библиотек, помогающих запускать 32-битные приложения в 64-битной среде.

» **Arch User Repository (AUR)** Неофициальный репозиторий пакетов, предоставленных пользователями. Технически Community и Community-testing являются частью AUR.

Документация по Arch

Первое, что нужно сделать еще до загрузки Arch и даже до записи его ISO-образа – зайти в вики по Arch на сайте http://wiki.archlinux.org. Для новичков и даже для тех, кому внове только Arch, «Руководство для начинающих» и «Официальное руководство по установке Arch Linux» обязательны к прочтению. Советуем держать их под рукой – открытыми в нетбуке или ноутбуке – во время установки системы на другой компьютер. Оба руководства исчерпывающе описывают все аспекты установки и дальнейшей настройки.

Известным источником огромного количества информации по Arch в целом является также вики. В разделе «Руководства [How-tos]» – более 641 страницы, и мы советуем ознакомиться с ними на досуге. Возможно, вы найдете всю информацию по установке своего любимого приложения в Arch на его специальной странице: https://wiki.archlinux.org/index.php/Category:HOWTOs_(English).

Почти так же, как вики, полезны форумы по Arch, на которых всегда кипит бурная деятельность: пользователи пытаются найти ответы на свои вопросы, и вам всегда смогут дать нужный совет. Касается ли ваш вопрос неработающей мыши и клавиатуры или настройки файла **xorg.conf**, на форумах найдется масса информации и для новичков, и для ветеранов: https://bbs.archlinux.org.

выберите текстовый редактор, с которым вам удобно работать. Возможные варианты – *Nano*, *Vi* и *Joe*. Если вы впервые сталкиваетесь с редакторами, запускаемыми из командной строки, советуем воспользоваться *Nano*, хотя *Vi* довольно удобен и к нему легко привыкнуть. Затем появится список всех основных файлов настройки. На данном этапе можно менять любой из них. И если вы хотите изменить список зеркал *Pacman* или описать точки монтирования в **/etc/fstab**, выберите файл из списка и нажмите Enter. Файл откроется в текстовом редакторе, выбранном на предыдущем шаге, и вы сможете легко изменить его желаемым образом. Покончив с файлами, прокрутите список вниз до Root-Password и нажмите Enter. Укажите пароль для пользователя root. Наконец, выберите пункт «Готово» (Done) и нажмите Enter для возврата в главное меню.

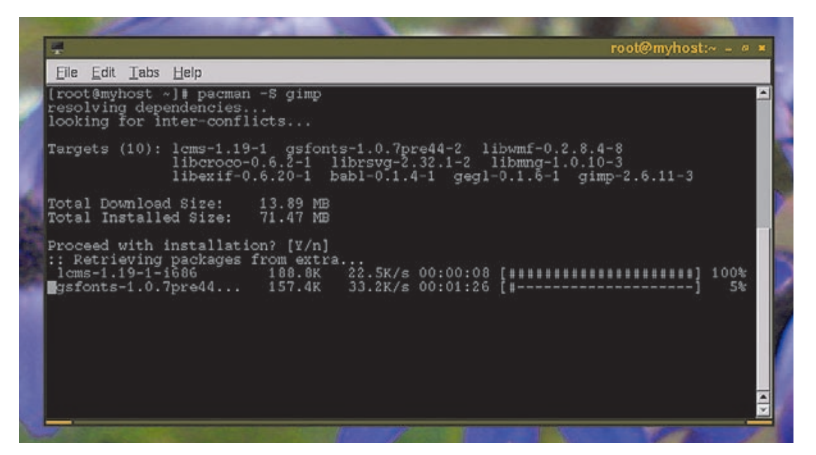
> «Если вы хотите вникнуть в Linux, лучшего учителя не найти.»

Последний этап – установка *Grub*. Его можно и не устанавливать: тогда измените настройки текущего и создайте в них запись для свежеустановленного Arch. Если вы решите установить *Grub*, установщик откроет для вас файле **menu.lst** в текстовом редакторе. Теперь выйдите из установщика и наберите **reboot** в командной строке, чтобы перезагрузить компьютер.

По завершении установки вы, несомненно, захотите настроить интернет-соединение. Если вы указали параметры своей сетевой карты в файле **/etc/rc.conf** в разделе «Сеть [Networking]» во время установки, сейчас у вас должно быть рабочее сетевое соединение. Команда **ifconfig -a** выведет список всех сетевых интерфейсов. Вы сможете установить статический IP-адрес интерфейса и шлюз по умолчанию с помощью утилит *ifconfig* и *route*, соответственно. Информацию о настройке беспроводной сети или работе с прокси-серверами можно найти в руководстве для начинающих в вики по Arch.

Содержимое файла **/etc/pacman.conf** по умолчанию подходят большинству пользователей. Репозитории Core и Extra уже подключены, и можно приступить к обновлению системы. Прежде всего измените файл **/etc/pacman.d/mirrorlist** и подключите зеркала, которые вы хотите использовать. Для этого просто удалите **#** в начале соответствующих строк. Затем выполните команду **pacman -Syy** для обновления списка пакетов. Эту команду нужно запускать каждый раз после изменения зеркал.

› ***Pacman* — среди лучших утилит управления пакетами, если не самый лучший.**



Обновите систему

Чтобы привести систему к самому свежему виду, выполните команду **pacman -Syu**. Она проведет обновление системы. Весьма вероятно, появится запрос на обновление *Pacman*. После этого снова запустите команду **pacman -Syu**. В зависимости от доступных обновлений может потребоваться изменение настроек, поэтому внимательно изучите вывод команды *Pacman* на наличие инструкций.

Теперь создайте нового пользователя командой **useradd**:

```
useradd -m -g [initial_group] -G [additional_groups] -s [login_shell] [username]
```

Загляните на man-страницу команды **useradd** и пробегитесь по этим параметрам: **-m** гарантирует создание в **/home** домашнего каталога пользователя, а **-g** указывает группу нового пользователя. Эта группа уже должна существовать. Можно указать группу **users**. После параметра **-G** можно указать несколько дополнительных групп, таких как **audio**, **storage** and **power**, чтобы пользователь мог контролировать эти параметры. Параметр **-s** задает оболочку по умолчанию. Здесь можно указать *Bash* или любую другую, но предварительно убедитесь, что оболочка установлена. Экипированная команда **useradd** обычно выглядит так:

```
useradd -m -g users -G audio,storage,power,wheel,video -s /bin/bash linuxlala
```

Теперь командой **passwd** задайте пароль для только что созданного пользователя.

Мы обсудили все аспекты настройки, и, наконец, настало время установить графическую среду. Для этого нужно установить как минимум *X Window System* и оконный менеджер. Простой оконный менеджер, такой как *Openbox* в комбинации с *X*, предоставит графический интерфейс, но при желании можно установить рабочий стол – Gnome, KDE, XFCE и др.

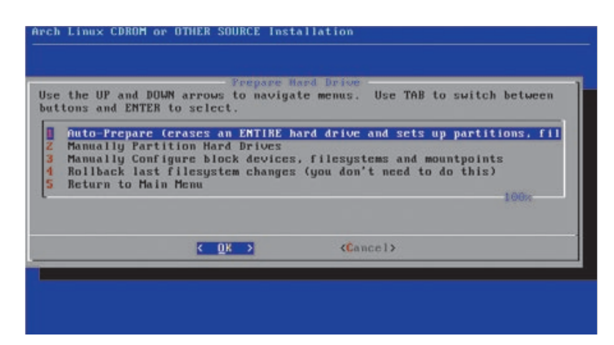
Оконный менеджер сам по себе предоставляет самый базовый интерфейс, так как в нем отсутствуют многие черты рабочего стола, например, иконки. Зато этот интерфейс обуславливает быструю систему, нетребовательную к ресурсам. В репозиториях Arch есть масса рабочих столов и оконных менеджеров; предлагаем вам тщательно выбрать себе соратника.

Благодаря высокой эффективности *Pacman* для установки *X* достаточно выполнить команду

```
pacman -S xorg-server xorg-xinit xorg-utils xorg-server-utils
```

Pacman автоматически определит зависимости и сообщит список требуемых к загрузке пакетов и общий объем загружаемых файлов. Для таких крупных установок решающее значение имеет правильный выбор зеркала. Если включены медленные или нера-

» **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.



› Для отмены последних изменений пользуйтесь опцией отката на странице «Подготовка жесткого диска».

ботающие зеркала, загрузка займет целую вечность, потому что для каждого пакета *Pacman* будет пытаться подключиться к этим неработающим зеркалам и перейдет к другим только тогда, когда соединение с первыми установить заведомо не удастся.

После установки этих пакетов и их зависимостей установите *Mesa* для поддержки 3D командой:

```
pacman -S mesa
```

Затем установите драйверы для видеокарты. Arch поддерживает ATI и Nvidia, а также старые встроенные чипы. В зависимости от вашей видеокарты может потребоваться установка драйвера VESA, ATI или Nvidia. Команда **lspci** поможет определить чипсет видеокарты. Теперь установите подходящий для вашего чипсета драйвер xf86-video. Обладателям видеокарт Nvidia подойдет драйвер Nouveau. Чтобы узнать, какой драйвер годится вам, загляните на страницы Nvidia и Nouveau в вики по Arch. Владельцам карт ATI также придется выбрать между свободным драйвером Radeon и проприетарным драйвером fglrx. Страница ATI в вики должна помочь с выбором.

Для связи приложений друг с другом используется настольная шина [Desktop Bus], или D-BUS. Это неотъемлемый компонент системы Linux, и ее нужно установить:

```
pacman -S dbus
```

Добавьте **dbus** в список известных демонов в файле **/etc/rc.conf**, и она будет запускаться автоматически при загрузке системы. Откройте файл **/etc/rc.conf** в любимом текстовом редакторе и прокрутите файл до строки **DAEMONS** в самом конце. Добавьте **dbus** в массив **DAEMONS** как указано ниже и сохраните файл:

Arch в VirtualBox

Arch легко устанавливается в *VirtualBox*, но, приступая к установке, некоторые вещи нужно знать. Хотя в целом информация в вики по Arch на https://wiki.archlinux.org самая свежая и уместная, страница *VirtualBox* – одна из немногих страниц, где содержатся неактуальные сведения. Инструкции в целом верны, но отсутствует несколько важных шагов.

Вдобавок нужно отметить, что общие инструкции по установке *X*, описанные на этих страницах, неприменимы. Можете воспользоваться инструкциями в вики (по ссылке https://wiki.archlinux.org/index.php/Arch_Linux_VirtualBox_Guest), только не устанавливайте гостевые дополнения [Guest Additions] из репозиториев. Также нужно установить *GTK 2* совместно с *X.org*.

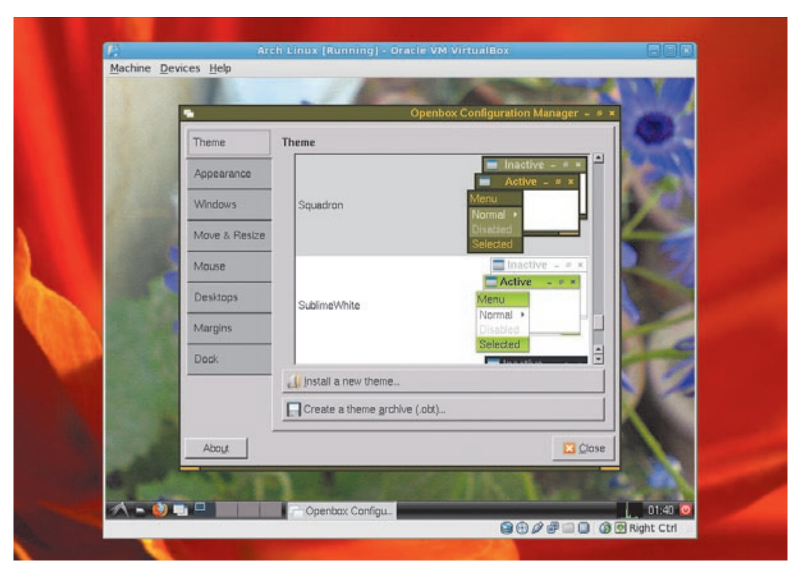
После этого вы готовы к установке гостевых дополнений с компакт-диска.Выберите Устройства > Установить гостевые дополнения [Devices > Install Guest Additions] и смонтируйте диск:

```
mkdir /mnt/disc
mount /dev/cdrom /mnt/disc
```

Затем запустите установочный скрипт:

```
cd /mnt/disc
./VBoxLinuxAdditions.run
```

Он установит в системе гостевые дополнения, и вы насладитесь ими с корректным разрешением монитора при переключении в полноэкранный режим.



› Arch в *VirtualBox* *4.0.4* в Fedora 14. С помощью *Openbox Configuration Manager* можно задать множество настроек.

```
DAEMONS=(syslog-ng dbus network netfs crond)
```

Рабочий стол на наших экранных снимках – LXDE, он основан на оконном менеджере *Openbox*. Для установки LXDE выполните команду **pacman -S lxde**; она установит рабочий стол, а также все зависимости и связанные пакеты. Можно установить Gnome, KDE или любой другой из предлагаемых рабочих столов. Дальнейшая настройка зависит от устанавливаемого рабочего стола, поэтому загляните в вики за соответствующими инструкциями по настройке.

После установки LXDE надо настроить *Openbox*. Если вы еще не создали пользователя, сделайте это сейчас. Зайдите в систему как этот новый пользователь и скопируйте три файла из каталога **/etc/xdg/openbox/** в каталог **~/.config/openbox/** нового пользователя:

```
cp /etc/xdg/openbox/* ~/.config/openbox
```

Вам также нужно установить *Gamin* – систему мониторинга файлов и каталогов – командой **pacman -S gamin**, а также утилиту *Obconf*, с помощью которой можно настроить *Openbox*, командой **pacman -S obconf**.

Чтобы загрузить свежеустановленный рабочий стол LXDE, скомандуйте

```
xinit /usr/bin/ck-launch-session startlxde
```

Также можно установить менеджер окон, такой как *GDM*, *KDM* или *Slim*, благодаря которому появится графическое окошко входа в систему. По умолчанию Arch выдаст приглашение для входа в терминале, а если вы хотите автоматически загружаться в *X*, загляните в раздел «Запуск *X* при загрузке» в вики.

Сделанный настройками

Из-за огромного объема действий по настройке Arch мы не могли осветить все аспекты его запуска в этой статье, но смогли показать, как с ним работать. Как мы убедились, Arch Linux – не для новичков, и если у вас нет времени и терпения на настройку дистрибутива в соответствии со своими нуждами, лучше с ним не связываться. Если вам нужен дистрибутив, с которым вы проверяли бы свою доску объявлений в Facebook уже через пару минут после установки, берите Ubuntu или Fedora.

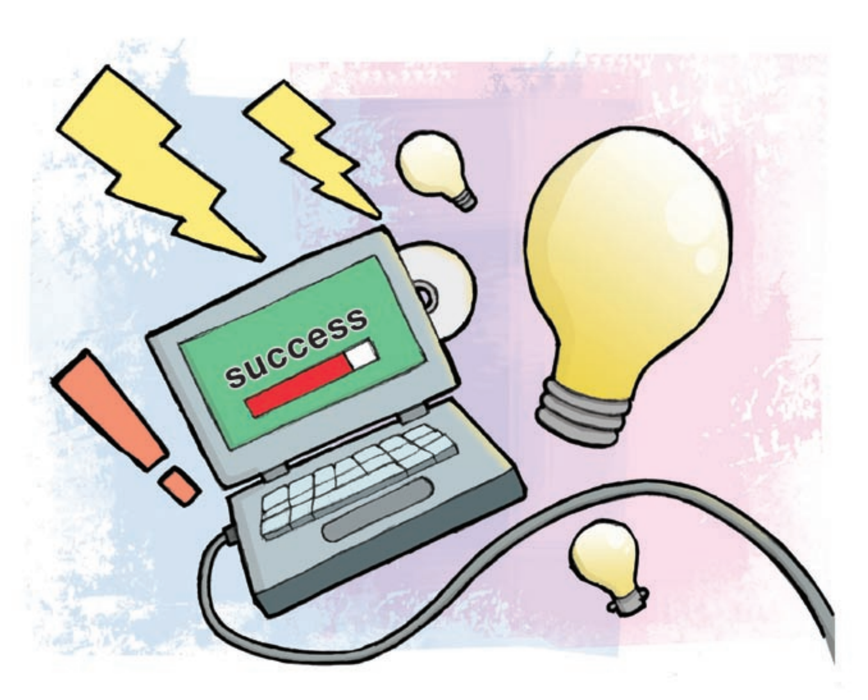
Но если вы интересуетесь Linux и хотите, наконец, вникнуть в его внутренности, если вы готовы узнавать, как все работает на самом деле и какие технологии лежат в основе современного компьютера, лучшего учителя, чем Arch, вам не найти. Его система управления пакетами, активные форумы и обширная документация в вики делают освоение Arch практически прогулкой в парке. С его активным сообществом вы никогда не будете одинокими на этом пути. LXF



Скрипт *rankmirror* поможет определить быстрые зеркала в файле **/etc/pacman.d/mirrorlist**. Для него потребуется установить пакеты **python** и **curl**.

KWin: Порулим

Коэн Вервлоесем не любит таскать окна и с ними валандаться, и рад был узнать, что последняя версия *KWin* поддерживает скрипты.





Наш эксперт

Коэн Вервлоесем пользуется свободными программами с 2000 года, и ему нравятся те, которые легко автоматизировать.

В KDE 4.6 оконный менеджер *KWin* получил новую прекрасную возможность: интерфейс скриптов. А значит, поведение окон можно настроить по своему вкусу. Пара примеров: все окна, кроме активного, можно затенить; можно сохранять расположение окна «поверх других», пока вы не развернете его и оно не будет считаться обычным окном. Также можно «склеить» два окна: при изменении размера одного размер второго будет автоматически меняться так, чтобы их контуры оставались вместе. Возможности безграничны. В этой статье мы покажем, что можно сделать с помощью скриптов *KWin*, и после ее прочтения вы сможете написать свои скрипты *Kwin* и оптимизировать поведение окон на своем рабочем столе.

Привет, мир

Прежде всего следует узнать, где *KWin* ищет скрипты – а это каталог **~/.kde4/share/apps/kwin/scripts/**. Все файлы с расширениями **.kwinscript**, **.kws** и **.kwinqs**, у которых установлен бит выполнения, будут распознаны *KWin* как файлы скриптов. По умолчанию при запуске *KWin* выполняет все обнаруженные им скрипты в этом каталоге. Проверим это. Создайте файл **helloworld.kwinscript** со следующим содержимым в каталоге **scripts**:

```
print("Hello world!");
```

Затем сделайте файл исполняемым:

```
chmod +x ~/.kde4/share/apps/kwin/scripts/helloworld.kwinscript
```

> «Добыть информацию об окнах можно и не дожидаясь события.»

Теперь перезагрузите *KWin* в *Konsole*; при этом будет отображаться вывод скрипта, а также некоторая отладочная информация:

```
kwin --replace &
```

Если вы не видите слов “Hello world!”, возможно, вы забыли сделать скрипт исполняемым или у вас более старая версия KDE. Также, возможно, вы допустили синтаксическую ошибку; но тогда *KWin* выдает сообщение об ошибке в консоли. Если все работает, можно двигаться дальше.

Различные объекты

Язык скриптов *KWin* – ECMAScript, стандартизированная версия JavaScript. Если у вас есть опыт программирования на JavaScript, то проблем с написанием скриптов *KWin* не будет. В дальнейшем мы будем считать, что вы знаете JavaScript, но объясним некоторые основные концепции для тех, у кого есть опыт программирования на других языках, чтобы они не отставали. Прежде всего, знайте, что существуют три вида объектов:

» **Инициализируемые [instantiable]** Они создаются с помощью конструкции **var foo = new Foo();**. Объектов такого типа можно создавать сколько угодно.

» **Одиночные [singleton]** Для этих объектов существует только один экземпляр заданного типа. Их нельзя создавать самим, но они предоставляются автоматически в скриптах *Kwin*.

» **Плавающие [floating]** Для этих объектов существует множество экземпляров заданного типа, но их нельзя создавать самим: они получаются в качестве возвращаемого значения метода.

Скрипты *KWin* управляются событиями: *KWin* генерирует определенные события, с которыми можно связать функции, вызываемые при наступлении события. Этот подход – ключ к изменению поведения программы согласно своим потребностям: нужно просто назначить собственные функции подходящим событиям. Следующий пример прояснит это:

```
workspace.clientMinimized.connect(function(client) {
print("Minimized: [" + client.caption() + "]");
});
```

Объект workspace здесь – одиночный; у него есть множество событий, инициируемых на уровне системы. Это основной источник событий в *KWin*, и вы будете пользоваться им во многих скриптах для инициализации взаимодействия с оконным менеджером. Например, событие **clientMinimized**, которым мы здесь воспользовались, возникает каждый раз при сворачивании окна. С помощью метода **connect** мы связываем с этим событием функцию. В данном случае функция безымянная; ей передается один параметр, клиент (представляющий собой окно, плавающий объект). В этой функции мы выводим сообщение о том, что клиент свернут, и применяем метод **caption()** объекта-клиента для получения заголовка окна. Сохраните этот скрипт, сделайте его исполняемым, перезагрузите *KWin* в *Konsole* и наблюдайте за выводом команды

окнами

при сворачивании окна. Безымянной функцией пользоваться не обязательно. Следующий код эквивалентен предыдущему:

```
function printCaption(client) {
print("Minimized: [" + client.caption() + "]");
}
workspace.clientMinimized.connect(printCaption);
```

Если функция у вас небольшая и используется только в одном месте, она может быть безымянной: меньше набора кода. Однако, когда функция разрастается или нужно вызывать ее в нескольких местах, лучше определить ее явно, чтобы код оставался понятным.

Методы и свойства

Добыть информацию об окнах можно и не дожидаясь наступления определенного события – для этого достаточно воспользоваться методами и свойствами. Метод – это функция, у которой есть возвращаемое значение. У объекта каждого типа есть несколько заданных методов. Например, для получения списка всех клиентов можно воспользоваться методом **getAllClients()** объекта **workspace**, который возвращает массив всех клиентов:

```
var clients = workspace.getAllClients();
for(var i = 0; i < clients.length; i++) {
print("[" + clients[i].caption() + "]");
}
```

Здесь мы создаем инициализируемый объект **clients** – в нем хранится массив всех клиентов. Затем в цикле **for** мы перебираем клиентов и выводим их заголовки. Размер массива мы можем получить с помощью свойства **length**.

Если вы захотите написать собственный скрипт, загляните в документацию по API скриптов *KWin* на http://rohanprabhu.com/files/kwsapi.html. На этой странице приведен список всех объектов, которые *KWin* распознает в своих скриптах – например, **workspace**, **toplevel**, **client**, **clientgroup**, **windowinfo** и **config**. Для каждого объекта в документации приведен список всех событий, методов и свойств с коротким описанием их назначения.

› **Когда вы заменяете *KWin* в *Konsole*, вы получаете всю информацию, которую вывели.**

Эта информация поможет вам понять, чего именно можно достичь с помощью скриптов. Нужно знать, что **toplevel** — родительский класс по отношению к **client**, поэтому все методы объектов **toplevel** также применимы к объектам клиентов.

Склеиваем окна вместе

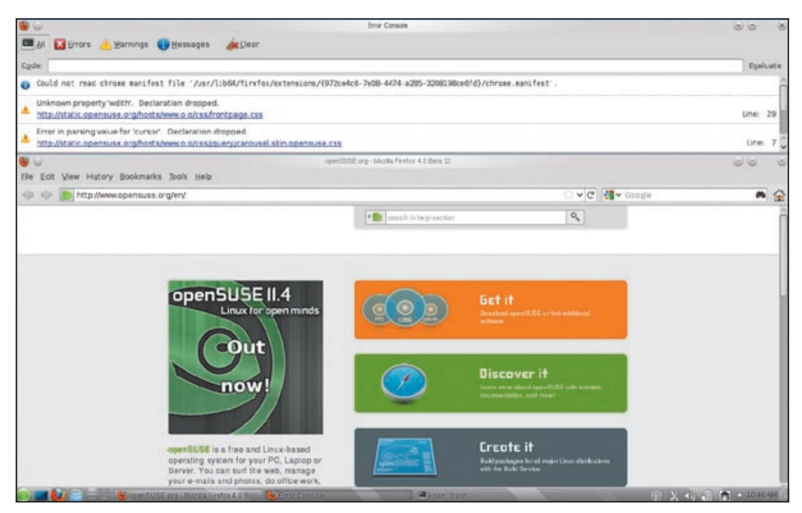
В принципе, ваших знаний уже достаточно для того, чтобы менять поведение окон по своему вкусу с помощью *KWin*. Чтобы придать вам вдохновения, приведем простой, но полезный пример: этот скрипт предназначен web-разработчикам, желающим, чтобы консоль ошибок всегда показывалась поверх главного окна *Firefox* »

Скрипты в других оконных менеджерах

KWin – не единственный оконный менеджер, позволяющий изменять поведение окон с помощью скриптов. Многие из так называемых «мозаичных» оконных менеджеров полностью поддерживают скрипты. Например, оконный менеджер *Awesome* допускает значительное расширение с помощью языка программирования Lua и обладает обширным и хорошо задокументированным API, с помощью которого можно тонко изменить поведение всех окон. Тот же подход использует *Xmonad*, но он написан на Haskell, чисто функциональном языке программирования, который не слишком просто освоить.

Существует несколько сторонних программ, которые «подцепляются» к оконному менеджеру и предоставляют вам контроль за местоположением и параметрами окон. Многие из них используют спецификацию расширенных подсказок оконного менеджера (Extended Window Manager Hints – EWMH), которой следует большинство популярных файловых менеджеров, в том числе *KWin*, *Metacity*, *Awesome*, *Xmonad* и *Openbox*. Например, существует программа *Devil's Pie*, которая считывает скрипты, позволяющие вам задавать правила для своих любимых приложений.

Команда **wmctrl** также предоставляет вам полный контроль над своими окнами, и она прекрасно подходит для использования в собственных скриптах оболочки. И если вам удобнее писать скрипты в *Bash*, чем программы на JavaScript, **wmctrl** определенно лучший выбор.



› Склейте окна навсегда с помощью скрипта *KWin*, и вам незачем будет насильственно совмещать их контуры.

и они видели бы в нем все ошибки и предупреждения. Вот как он выглядит:

```
var main_present = 0;
var console_present = 0;
var console_height = 200;
var main;
var console;
function viewportHeight() {
  return (workspace.dimensions().h)/(workspace.
desktopGridSize().h);
}
function viewportWidth() {
  return (workspace.dimensions().w)/(workspace.
desktopGridSize().w);
}
function adjustMain() {
  var geom = console.geometry();
  main.move(0, geom.height);
  main.resize(viewportWidth(), viewportHeight() - geom.height);
}
workspace.clientAdded.connect(function(client) {
   var z = client.getWindowInfo();
    if(z.windowClassClass == "Firefox") {
       if(client.caption() == "Error Console") {
           console = client;
           console_present = 1;
           console.move(0, 0);
           console.resize(viewportWidth(), console_height);
           if(main_present == 1) {
             adjustMain();
           }
         } else if(z.windowClassName == "Navigator") {
          main = client;
          main_present = 1;
          if(console_present == 1) {
            adjustMain();
          }
       }
     }
});
```

Скорая помощь

При проверке скрипта *KWin* для просмотра значений переменных, особенно заголовка клиента, класса **windowclass** и имени окна, в целях отладки пользуйтесь командой **print**.

Код начинается с объявления некоторых переменных: мы объявляем переменные для хранения главного окна и консоли окна ошибок. Мы также объявляем переменные **main** и **console**, которые будут содержать ссылки на клиентов главного окна и окна консоли ошибок. Затем мы определяем две вспомогательных функции для получения ширины и высоты экрана рабочего стола. В функции **adjustMain** мы изменяем ширину главного окна так, чтобы оно располагалось прямо под окном консоли ошибок.

Большая часть кода – безымянная функция, которая будет вызываться при каждом при наступлении события **clientAdded**. Сначала мы записываем информацию об окне в переменную **z**. **Windowinfo** – объект, дающий доступ к обширной низкоуровневой информации *X Window* об окне. В данном случае нас интересуют класс **windowclass** и имя окна: с их помощью мы определим, является ли клиентом *Firefox* и какое это окно: у каждого окна браузера его классом **windowclass** является **Firefox**. Кроме того, наш код отличает главное окно от консоли ошибок по значению имени **windowclass**: **Navigator** и **Error Console** соответственно.

> «Наш код отличает главное окно от консоли ошибок.»

Таким образом, при каждом добавлении нового клиента мы проверяем, является ли он консолью ошибок *Firefox* или главным окном браузера. В первом случае мы располагаем окно в левом верхнем углу экрана и задаем ему ширину по умолчанию в 200 пикселей. Затем мы смотрим, есть ли главное окно, и если есть, также изменяем его размер. С другой стороны, если только что добавленный клиент – главное окно, мы также меняем его размер, если консоль уже есть.

События для конкретных клиентов

Впрочем, можно сделать и лучше: нельзя ли нам автоматически изменять размер главного окна при изменении размера окна консоли, чтобы они всегда были «приклеены» друг к другу? Для этого свяжем безымянную функцию с событием **clientMoved**, генерируемым для консоли. Добавьте следующие строки в конец кода, который выполняется при добавлении окна консоли (до **} else if...**):

```
console.clientMoved.connect(function() {
if(main_present == 1) {
adjustMain();
}
});
```

Вкладки

Здесь мы работали только с отдельными окнами, но с помощью объекта **clientgroup** можно получить доступ к вкладкам. Чтобы создать новую группу вкладок, просто объявите переменную **var group = new ClientGroup(client);** где **client** – первое окно, которое вы хотите добавить. Остальные окна можно добавить с помощью функции **clientgroup.add**, удалить – с помощью метода **clientgroup.remove**, удалить все – с помощью метода **clientgroup.removeAll**.

У объекта **clientgroup** есть множество интересных методов для работы с клиентами в группе. Например: **clientgroup.clients** возвращает массив всех клиентов, являющихся членами группы; с помощью **clientgroup.contains(client)** можно проверить, является ли конкретный клиент членом группы.

Clientgroup.indexOf(client) возвращает положение клиента в группе вкладок, а пользуясь **clientgroup.move**, можно переместить окно клиента в другое положение. Таким образом, с помощью объекта **clientgroup** можно написать собственный оконный менеджер с вкладками поверх KWin.

» **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.

Разница между тем, что мы делали раньше, и тем, что делаем сейчас, в том, что **clientMoved** — событие, генерируемое не на уровне системы для объекта **workspace** с клиентом в качестве параметра, а для конкретного клиента; в данном случае, для окна консоли. И когда мы связываем безымянную функцию с этим событием, функции не нужен клиент в качестве параметра, потому что мы уже знаем, что это за клиент. В результате при каждом перемещении (в том числе и при изменении размеров) окна консоли положение главного окна изменяется соответствующим образом.

Однако иногда было бы интересно распознать клиента внутри функции. В данном случае нужно указать объект следующим образом:

```
console.clientMoved.connect(client, function() {
  if(main_present == 1) {
  adjustMain();
  print(client.caption() + " is moved");
  }
});
```

Таким образом наша безымянная функция получает доступ к клиенту, генерирующему событие, в данном случае – через переменную клиента. Здесь нам это не нужно, поскольку мы знаем, что функция вызывается только для окна консоли, но если функция будет вызываться различными клиентами, нам понадобится переменная, чтобы их различать.

Файл настройки скрипта

Если вы хотите задавать для файла скрипта настройки, просто создайте файл с тем же именем, но с расширением **.kwscfg** в каталоге **~/.kde4/share/apps/kwin/scripts/**. Этот файл должен иметь формат INI-файла с парами **ключ-значение** в следующем виде:

```
consoleHeight=200
```

Как вы видите, в этом файле настройки мы задаем ширину по умолчанию для окна консоли из нашего скрипта «склеивания» окон. Теперь нам остается только воспользоваться этим значением в скрипте, изменив объявление переменной **console_height** следующим образом:

```
var console_height = config.get("consoleHeight");
```

Здесь используется одиночный объект **config**, и в дальнейшем, если в скрипте нужно использовать некоторые заданные значения, лучше задавать их в файле настройки и получать в своем скрипте функцией **config.get**. Вы всегда можете захотеть изменить эти значения, и проще и менее рискованно сделать это в файле настройки, чем вторгаясь в код.

> «В следующих версиях непременно появятся новые функции.»

Если в файле настройки больше одной пары **ключ-значение**, можно получить их так же, как в этом примере, но также можно вызвать **config.get()** без параметров и получить все ключи и значения в ассоциативном массиве. Или с помощью вызова **config.get("key1", "key2", "key3")** можно получить ассоциативный массив заданных ключей и их значений.

Если в файле настройки нужно проверить только наличие ключа, воспользуйтесь вызовом **config.exists("key")**. При отладке можно воспользоваться методом **config.loaded**, который вернет true, если файл настройки был найден и загружен, и false в противном случае. Учтите, что у каждого скрипта *KWin* есть доступ только к своему файлу настройки.

Что дальше?

Поддержка скриптов в *KWin* все еще находится на начальном уровне, и это ясно из отсутствия многих событий, методов и свойств, способных представлять интерес. Например, у объекта **client** есть методы **isShade()** и **isShadeable()** для определения, затенен ли клиент и можно ли его затенить, но нет метода **setShade()**, чтобы изменить состояние затененности клиента. У объекта клиента также есть метод **isMaximizable()** и событие **maximizeSet**, которое генерируется при разворачивании окна, но (все еще) нельзя программно развернуть клиента, например, методом **setMaximize()**.

Это означает что на практике при появлении той или иной идеи вы часто не сможете ее воплотить, потому что скрипты не поддерживают необходимых вам функций. Но не забывайте, что это первая версия KDE с поддержкой *KWin*. В следующих версиях непременно появятся новые функции, и вы в конце концов сможете автоматизировать почти все свои задачи управления окнами в KDE. Управление событиями и тот факт, что скрипты *KWin* можно писать на обычном JavaScript, делают их доступными большому количеству пользователей.

Рохан Прабху [Rohan Prabhu], главный разработчик поддержки скриптов *KWin*, написал руководство, доступное в его блоге: http://rohanprabhu.com/?p=116. В нем он пишет скрипт, сохраняющий положение окна «поверх всех окон», пока оно не развернуто. Когда окно разворачиваются, оно становится обычным, и его можно накрыть другими окнами, когда оно развернуто. Когда окно неразвернуто, скрипт снова автоматически располагает его «поверх всех окон».

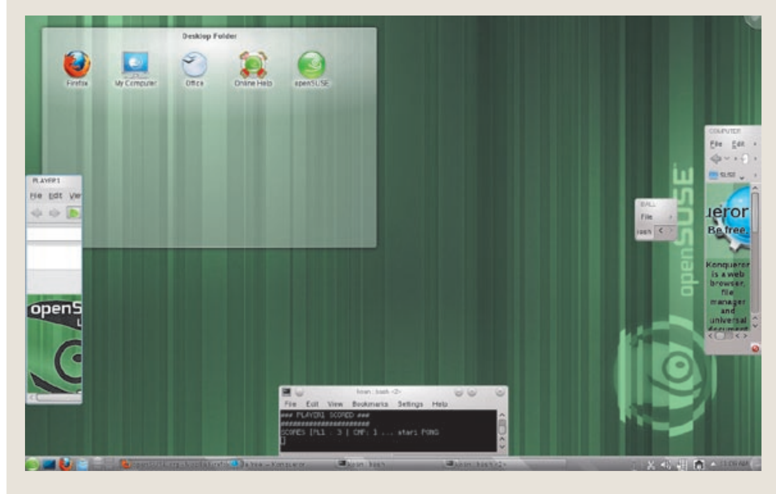
Прабху регулярно пишет о различных аспектах написания скриптов KWin в своем блоге; ознакомиться с этими записями можно по ссылке http://rohanprabhu.com/?tag=kwin_scripting. С его блогом, документацией по API и вдохновением, которое, надеюсь, у вас появилось, у вас есть все необходимое для успешного программирования... LXF

Пинг-понг с вашими окнами

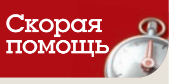
Для иллюстрации возможностей скриптов *KWin* Рохан Прабху даже написал пинг-понг в скрипте *KWin*. Не то чтобы было сильно удобно, когда окна летают по экрану, но это хороший пример, с которым вам непременно стоит познакомиться.

Со страницы http://rohanprabhu.com/?p=56, установите скрипт в каталог скриптов *KWin* и сделайте его исполняемым, затем перезагрузите *KWin* в *Konsole*, после чего скрипт в *Konsole* попросит вас выбрать клавиши управления.

Например, можно устроить войну браузеров, выбрав своей ракеткой *Firefox*, а ракеткой компьютера – *Chrome* или *Konqueror*. Затем выберите другое окно в качестве мяча, а в окне *Konsole* вы увидите счет матча.



› **Война браузеров в виде пинг-понга на вашем рабочем столе.**



При проверке сложных скриптов *KWin* проявляйте осторожность. Например, если вы сделаете ошибку в поведении активных окон, вы даже не сможете вернуться в *Konsole*, чтобы перезагрузить программу.

Erlang: Базовые

Базовыми сущностями языка функционального программирования, естественно, являются функции. **Андрей Ушаков** начинает новую серию уроков...





Наш эксперт

Андрей Ушаков Активно приближает тот день, когда функциональные языки станут мейнстримом.

В предыдущих выпусках (**LXF 143**, **144**) была напечатана статья, посвященная введению в язык программирования Erlang. В новом цикле статей я продолжу рассказ о языке Erlang и концепциях функционального программирования. Данная статья посвящена одной из фундаментальных сущностей вообще всех языков программирования (в том числе и Erlang) – функциям.

Во многих языках программирования функции не являются типом данных. Это означает, что я не могу объявить переменную и присвоить ей функцию, либо передать функцию как аргумент вызова другой функции. Конечно, не все так плохо, но для работы с функциями как с типами данных приходится совершать дополнительные действия. Так, например, в C++ для этого мы вводим указатель на функцию, либо вместо функций используем функторы; в Java используем типы-обертки (например, анонимные классы) либо ссылки на метаданные. Понятно, что хочется работать с функциями, как с другими типами данных: было бы удобно иметь возможность легко объявить переменную и присвоить ей в качестве значения функцию, либо передать одну функцию в качестве аргумента другой. Язык Erlang, так же как и другие функциональные языки программирования, это позволяет. Давайте поговорим о функциях более подробно.

> «Хочется работать с функциями как с другими типами данных.»

Все функции всегда определены в модулях. Объявление модуля – это всегда первая строка в файле. Модуль объявляется при помощи директивы **module**, содержащей имя модуля. Имя модуля есть атом (и поэтому должно начинаться с маленькой буквы), совпадающий с именем файла без расширения. Все функции в модуле делятся на экспортируемые и неэкспортируемые. По умолчанию функция является неэкспортируемой и будет видна только внутри модуля. Чтобы сделать ее экспортируемой, необходимо ее сигнатуру (сигнатура функции – это имя функции, после которого идет знак '/' и арность функции) прописать в директиве **export**. Директива **export** содержит список сигнатур экспортируемых функций. Например, объявление модуля и экспортируемых функций может выглядеть следующим образом (имя файла, соответственно, **example1.erl**):

```
-module(example1).
-export([func1/1, func2/2, func3/0]).
```

Использовать функции из того же модуля, в котором они объявлены, просто: достаточно обратиться к ним по имени (и передать соответствующий список аргументов). А что же с функциями из других модулей (понятно, что речь идет только об экспортируемых функциях)? У нас есть два варианта решения этого вопроса (так же как и в большинстве языков программирования): использовать полное имя функции либо импортировать функции из другого модуля. Полное имя функции – это имя модуля, в котором функция определена, после которого идут знак '**:**' и имя функции, заданное при определении. Например, обращение к функции **seq**, определенной в модуле **lists**, будет выглядеть следующим образом: **lists:seq(1, 10).** Импорт функций, с другой стороны, позволяет использовать функции из других модулей по их имени, заданному при определении. Директива импорта выглядит следующим образом:

```
-import(ModuleName, FuncList).
```

Здесь **ModuleName** – имя модуля, из которого производится импорт; **FuncList** – список импортируемых функций. Например, импорт функций **seq/2** и **seq/3** из модуля **lists** будет выглядеть следующим образом:

```
-import(lists, [seq/2, seq/3]).
```

Так же, как и в большинстве языков программирования, в **Erlang** есть автоматически импортируемые функции. Эти функции называются BIF и импортируются они (не все) из модуля **erlang**.

Обратим свой взор теперь к объявлению функции. А точнее, на объявление нескольких вариантов одной и той же функции. Как вы помните из предыдущих статей, при вызове функции поиск подходящего варианта осуществляется при помощи двух механизмов (которые могут работать совместно в одном варианте): операции соответствия шаблону [pattern-matching] и выражения охраны [guards]. И если с операцией соответствия шаблону все достаточно просто, то с выражением охраны ситуация более интересная. Выражение охраны – это булевское выражение (на самом деле выражение охраны может возвращать любой атом, но истинным значением бу-

сущности

дет значение **true**), и вопрос заключается в том, какие операции я как разработчик могу использовать. Например, вправе ли я написать объявление такого варианта функции:

```
calculate(X) when math:sin(X) > math:cos(X) → … ;
```

Из документации видно, что в выражениях охраны могут появляться только следующие операции: атомы, операции сравнения, арифметические выражения, логические выражения и ограниченный набор BIF'ов. В этот набор входят все функции, проверяющие тип аргумента: **is_*/1** (например, **is_atom/1**), **abs/1**, **bit_size/1**, **byte_size/1**, **element/2**, **float/1**, **hd/1**, **length/1**, **node/0**, **node/1**, **round/1**, **self/0**, **size/1**, **tl/1**, **trunc/1**, **tuple_size/1**. И ответ на приведенный выше вопрос будет следующий: мы не вправе написать подобное объявление варианта функции – его не пропустит компилятор.

Зачем сделано такое ограничение на выражения охраны? Затем, чтобы гарантировать, что выражения охраны свободны от побочных эффектов. Что же делать, если мне необходимо выбрать вариант функции в зависимости от более сложного условия, чем позволяет задать разрешенный набор операций в выражениях охраны (как в приведенном примере)?

Ответ достаточно очевиден: использовать одно из выражений **if** или **case**. Рассмотрим приведенный выше пример: пусть **calculateWhenX/1** и **calculateWhenY/1** представляют две ветви выполнения (то, что мы хотели записать как два варианта функции), тогда функция **calculate/1** будет иметь следующий вид:

```
calculate(X) →
        if
                math:sin(X) > math:cos(X) →
calculateWhenX(X);
                true → calculateWhenY(X)
        end.
```

И последнее про выражения охраны: выражение охраны на самом деле может быть списком выражений, использующим в качестве разделителя либо символ ';', либо символ ','. В первом случае выражение охраны истинно, если истинно хотя бы одно выражение из списка; во втором случае – если истинны все выражения.

Перейдем к рассмотрению рекурсии. Рекурсия – это возможность функции вызывать саму себя. Реализуется она через стек, и не случайно, что при большой глубине рекурсивных вызовов наступает переполнение стека. Есть рекурсия и в Erlang, но помимо обычной рекурсии в Erlang присутствуют рекурсивные вызовы специального типа – хвостовая рекурсия. Если в результате выполнения тела функции последней операцией будет рекурсивный вызов самой себя, то такой рекурсивный вызов называется прямой хвостовой рекурсией. Возможна ситуация, когда в результате выполнения функции **A** последней операцией будет вызов функции **B**, а в результате выполнения функции **B** последней операцией будет вызов функции **A**. И в этом случае подобный рекурсивный вызов будет являть хвостовую рекурсию, только подобная рекурсия называется непрямой хвостовой рекурсией.

Особенность хвостовой рекурсии в том, что компилятор умеет обрабатывать такой случай нерекурсивным образом и, следовательно, хвостовая рекурсия может быть сколь угодно глубокой и при этом не вызовет переполнения стека. Рекурсия важна при реализации рекурсивных алгоритмов и/или при работе с рекурсивными структурами данных (например, с деревьями). В Erlang рекурсия (точнее, хвостовая рекурсия) еще важна и потому, что это единственный механизм построения циклических структур управления (циклических алгоритмов). Так, например, в Erlang отсутствует цикл **for**; предположим, что нам необходима структура управления, эмулирующая этот цикл (для простоты предположим, что нам необходимо эмулировать цикл **for**, который просто выполняется заданное число раз). Тогда мы можем эмулировать цикл **for** следующим образом:

```
for(Start, Count, Func, Acc) → for_impl(Start, 0, Count, Func, Acc).
for_impl(_, Index, Count, _, Acc) when Index >= Count → Acc;
for_impl(Start, Index, Count, Func, Acc) →
        for_impl(Start, Index+1, Count, Func, Func(Start + Index,
Acc)).
```

и далее, мы можем создать список из чисел от 1 до 10 следующим образом:

```
for(1, 10, fun(Number, Acc) → Acc ++ [Number] end, []).
```

В приведенном выше примере я использовал объявление анонимной функции. Давайте рассмотрим их более подробно. Анонимная функция (или лямбда) – это безымянная функция, определяемая локально по месту использования. В чем преимущества определения по месту таких безымянных функций? Такое определение более наглядно, чем определение отдельной функции, когда объявляемая функция небольшая. Если определение анонимной функции используется только в одном месте, то такой способ не «загрязняет» исходный код еще одним определением функции и не влияет на операцию поиска функции по сигнатуре, выполняемую компилятором. Определение анонимной функции в общем случае имеет следующий вид:

```
fun
        (Pattern11, …, Pattern1N) [when Guard1] → Body1;
        … ;
        (PatternK1, …, PatternKN) [when GuardK] → BodyK
end
```

Из общего определения видно, что и в анонимной функции мы можем задавать несколько ее вариантов, выбор которых может происходить как за счет операции соответствия шаблону, так и за счет выражений охраны. Если анонимная функция нужна только для передачи вызова обычной, неанонимной функции, то вместо написания подобного кода:

```
fun(X1, …, XN) → module:func_name(X1, …, XN) end
```

мы можем написать более коротко: **fun module:func_name/N** (ну или **fun func_name/N** – для функции из текущего модуля). Такая короткая запись обычно используется, чтобы создать ссылку на существующую неанонимную функцию.

Перейдем теперь к следующему понятию функционального программирования – функциям высшего порядка. Функции высшего порядка отличаются от обычных функций только тем, что они принимают в качестве аргумента функцию, либо их возвращаемое значение есть функция (либо и то, и другое). Достаточно очевидны случаи, когда нам нужно, чтобы одна функция »

принимала другую в качестве параметра. Например, если вы разрабатываете библиотеку для численного интегрирования, то вам нужно передавать в функцию интегрирования – в качестве одного из параметров – функцию, по которой считается определенный интеграл. Несложно придумать еще массу примеров, когда будет необходимость в параметризации одной функции другой. Более того, взглянув на стандартную библиотеку, поставляемую с **Erlang**, можно увидеть массу функций, которые ожидают, что один (или несколько) из параметров будет другой функцией. Придумать пример функции, у которой возвращаемое значение – другая функция, несколько более сложно, но тоже возможно. Таким примером может быть функция-фабрика (паттерн «фабрика»), создающая и возвращающая другую функцию (подобную функцию-фабрику мы создадим в примере ниже).

С функциями высшего порядка тесно связано еще одно понятие функционального программирования – карринг [currying] (или каррирование функции). Карринг – это преобразование функции от пары аргументов в функцию, берущую свои аргументы по одному. Другими словами, это преобразование функции от пары аргументов в функцию от одного аргумента, возвращающую функцию от одного аргумента. Возникает вопрос: а зачем вообще это нужно? Для частичного задания аргументов функции прямо сейчас! Например, у нас есть функция, позволяющая вести поиск одного фрагмента текста в другом. Очевидно, что это функция двух аргументов (минимум двух аргументов). Предположим, что набор фрагментов текста, которые мы ищем, заранее предопределен, а фрагмент текста, в котором ведется поиск, заранее неизвестен. В таком случае было бы удобно иметь предопределенный набор функций от одного параметра, для поиска предопределенного фрагмента текста в заданном. Но создавать новую функцию для каждого предопределенного фрагмента текста (по которому будет вестись поиск) будет неправильно. И вот тут вступает в дело карринг – мы преобразуем функцию от двух аргументов в функцию, берущую свои аргументы по одному, и задаем один из аргументов. Пусть функция поиска одного фрагмента в другом называется **search_text/2**, тогда карринг функции будет выглядеть следующим образом:

> **«Мы можем преобразовать арифметическое выражение в дерево.»**

```
SearchFun = fun(Search) → fun(Source) → search_text(Source, Search) end end,
SearchFragment1 = SearchFun("fragment1"),
```

Переменная **SearchFun** содержит ссылку на каррированную функцию **search_text/2** (т. е. ссылку на функцию одного аргумента, возвращающую функцию одного аргумента), а переменная **SearchFragment1** просто содержит ссылку на функцию одного аргумента, которая в итоге будет искать в заданном фрагменте текста строку **"fragment1"**.

Настало время применить полученные знания на практике: давайте напишем простой парсер арифметических выражений. На вход он будет получать арифметическое выражение в виде строки, которая может содержать целочисленные константы, имена переменных и все арифметические действия. На выход он будет выдавать функцию одного аргумента, содержащую распарсенное выражение. В дальнейшем, передавая полученной функции список пар «имя переменной – значение» (список кортежей, в котором первый элемент – имя переменной в виде строки, второй элемент – значение этой переменной), мы можем вычислять значение этого выражения для конкретных значений переменных. Для простоты наш парсер будет содержать следующие допущения: считаем, что арифметическое выражение не содержит скобки, знаки перед константами и переменными, исходную строку не чистим от пробельных символов, не вводим полноценную обработку ошибок.

Итак, начнем. Первое, что мы должны сделать – это разбить полученную строку на лексемы. В нашем случае лексемы могут быть следующие: целочисленная константа, имя переменной, знак арифметического действия. Знак арифметического действия, помимо того, что является лексемой, является также и разделителем, разбивающим выражение на лексемы (т. к. знаки перед константами и переменными не поддерживаются). Например, строка **2+a** разобьется на следующие лексемы: **2**, **+**, **a**. Для разбития исходной строки на лексемы служит функции **get_tokens/2** и **get_tokens_impl/2**:

```
get_tokens(InputStr, Delimiters) ->
        get_tokens_impl(InputStr, Delimiters, []).
get_tokens_impl([], _, TokenList) -> lists:reverse(TokenList);
get_tokens_impl(InputStr, Delimiters, TokenList) ->
        {Token, Rest} = lists:splitwith(fun(Char) -> not
lists:member(Char, Delimiters) end, InputStr),
        if
                Rest /= [] ->
                        [DelimiterChar | NextRest] = Rest,
                        get_tokens_impl(NextRest,
Delimiters, [[DelimiterChar]] ++ [Token] ++ TokenList);
                Rest == [] ->
                        get_tokens_impl([], Delimiters,
[Token]++TokenList)
        end.
```

После разбиения строки на лексемы наступает стадия парсинга. В результате парсинга мы должны получить некую структуру данных, позволяющую вычислять значение функции при заданных значениях переменной. Что же должна представлять собой эта структура данных, с учетом того факта, что в результате она должна быть обернута в функцию от одного аргумента? Рассмотрим для начала константу. Константу можно представить в виде функции от двух аргументов: списка пар «имя переменной – значение» и значения константы, возвращающей значение константы. Во время парсинга, при помощи карринга и частичного задания аргументов (задавая значение константы), мы можем преобразовать эту функцию в функцию от одного аргумента. Рассмотрим теперь переменную. Переменную можно представить в виде функции от двух аргументов: списка пар «имя переменной – значение» и имени переменной, возвращающей значение переменной по ее имени. Точно так же, как и в случае константы, во время парсинга мы можем преобразовать эту функцию в функцию от одного аргумента.

Рассмотрим, наконец, какую-либо бинарную операцию – например, сложение. Подобную операцию можно представить в виде функции трех аргументов: списка пар «имя переменной – значение», левого операнда от одного аргумента и правого операнда от одного аргумента, возвращающую результат выполнения операции. И точно так же, как в случае константы и переменной, мы можем преобразовать эту функцию в функцию от одного аргумента. Действуя подобным образом, мы можем преобразовать все арифметическое выражение в дерево из преобразованных (при помощи карринга и частичного задания аргументов) функций от одного аргумента. Функция **constant_fun/2** служит для представления константы, функция **variable_fun/2** – для представления переменной, функции **addition_fun/3**, **subtraction_fun/3**,

multiplication_fun/3, **division_fun/3** – для представления арифметических действий. В статье приводим объявление только для функции **addition_fun/3**, т.к. объявление остальных функций для арифметических действий аналогичное:

```
constant_fun(_, Value) -> Value.
variable_fun(VarList, Name) ->
        FindResult = lists:keyfind(Name, 1, VarList),
        if
                FindResult == false -> erlang:error({variable_
not_found, Name});
                true -> element(2, FindResult)
        end.
addition_fun(VarList, LeftMember, RightMember) ->
        LeftMember(VarList) + RightMember(VarList).
```

Следующий шаг – преобразование этих функций в функции от одного аргумента при помощи карринга и частичного задания аргументов. Это происходит при создании операндов (функции **build_operand/1**, **build_operand_impl/2**) и операций (функция **build_operator/3**):

```
build_operand(Operand) ->
        build_operand_impl(Operand,
string:to_integer(Operand)).
build_operand_impl(Operand, {error, _}) →
        fun(VarList) -> variable_fun(VarList, Operand) end;
build_operand_impl(_, {Int, []}) →
        fun(VarList) -> constant_fun(VarList, Int) end.
build_operator("*", LeftOperand, RightOperand) ->
        fun(VarList) -> multiplication_fun(VarList, LeftOperand,
RightOperand) end;
build_operator("/", LeftOperand, RightOperand) ->
        fun(VarList) -> division_fun(VarList, LeftOperand,
RightOperand) end;
build_operator("+", LeftOperand, RightOperand) ->
        fun(VarList) -> addition_fun(VarList, LeftOperand,
RightOperand) end;
build_operator("-", LeftOperand, RightOperand) ->
                fun(VarList) -> subtraction_fun(VarList,
LeftOperand, RightOperand) end.
```

Следует отметить, что создавать операнды нужно только для констант и переменных; распарсенная часть выражения и так является операндом для текущей операции.

Далее наступает самое интересное: преобразование списка лексем в дерево функции. Если бы все операции имели одинаковый приоритет, то подобное преобразование было бы тривиальной операцией: иди себе просто по списку лексем и преобразовывай по мере прохождения. В нашем случае все немного сложнее – приоритет операции имеют разный. Пусть мы идем по списку лексем и преобразуем его в дерево по мере прохождения. Пусть есть уже преобразованная часть (левый операнд), и текущая операция – низкоприоритетная. Тогда наши действия зависят от того, какая операция стоит после правого операнда. Возможны три случая: после правого операнда больше операций нет (конец списка); после правого операнда стоит низкоприоритетная операция; и после правого операнда стоит высокоприоритетная операция. В первых двух случаях все хорошо, и мы можем преобразовать текущую операцию в дерево функций, связывающее левый и правый операнды. В третьем же случае мы запоминаем левый операнд и текущую операцию и начинаем строить новое дерево, начиная с высокоприоритетной операции. Когда мы встретим низкоприоритетную операцию или конец списка, мы объединяем два дерева функций в одно при помощи запомненной операции (левый операнд – запомненное дерево, правый операнд – новое дерево) и идем далее. Преобразованием в дерево функций занимаются функции **build_fun/1**, **build_fun_impl/2** и **process_operands/4**:

```
build_fun(TokenList) ->
[LeftOperand | Rest] = TokenList,
        build_fun_impl(Rest, build_operand(LeftOperand)).
build_fun_impl([], {PrevOperand, Operator, LeftOperand}) ->
        build_operator(Operator, PrevOperand, LeftOperand);
build_fun_impl([], LeftOperand) -> LeftOperand;
build_fun_impl(TokenList, {PrevOperand, PrevOperator,
LeftOperand}) ->
        [Operator | Rest] = TokenList,
        [RightOperand | NextRest] = Rest,
        CurrentOperand = build_operator(Operator,
LeftOperand, build_operand(RightOperand)),
        process_operands(PrevOperand, PrevOperator,
CurrentOperand, NextRest);
build_fun_impl(TokenList, LeftOperand) ->
        [Operator | Rest] = TokenList,
        [RightOperand | NextRest] = Rest,
        process_operands(LeftOperand, Operator, build_
operand(RightOperand), NextRest).
process_operands(LeftOperand, Operator, RightOperand, []) ->
        build_fun_impl([], build_operator(Operator,
LeftOperand, RightOperand));
process_operands(LeftOperand, Operator, RightOperand,
TokenList) ->
        NextOperator = hd(hd(TokenList)),
        IsPriorityOperator = lists:member(NextOperator, "*/"),
        if
                IsPriorityOperator -> build_fun_
impl(TokenList, {LeftOperand, Operator, RightOperand});
                true -> build_fun_impl(TokenList, build_
operator(Operator, LeftOperand, RightOperand))
        end.
```

Ну что же, осталось совсем немного: объявление модуля, списка экспортируемых функций и экспортируемой функции **parse/1**:

```
-module(arith_parse).
-export([parse/1]).
parse(InputStr) ->
        build_fun(get_tokens(InputStr, "*/+-")).
```

Сохраняем исходный код в файле с именем **arith_parse.erl**, запускаем среду выполнения Erlang. В консоли Erlang запускаем компиляцию: командой **c(arith_parse).**, после чего можно приступать к тестированию. Сначала получаем распарсенное дерево функции:

```
F1 = arith_parse("3-2*a+b/4*3-1").
```

После, мы можем посчитать значение арифметического выражения для конкретных значений переменных:

```
F1([{"a", 9}, {"b", 13}]).
```

Для значений переменных a = 9 и b = 13 получаем ответ –6.25, правильность которого легко проверить вручную

В порядке заключения: способы использования функций в функциональных языках и похожи, и не похожи на те, к которым мы привыкли. Мы привыкли, что функции – это контейнеры для исполняемого кода, и если мы хотим их использовать как-то иначе, то вынуждены делать дополнительные и не всегда удобные действия. В функциональных языках, как мы увидели, все гораздо проще: функции одновременно являются и контейнерами для кода, и полноценным типом данных. Более того, некоторые техники делают использование функций более удобным и гибким, по сравнению с использованием в императивных языках. А в следующей статье мы рассмотрим другую базовую сущность функциональных языков (и Erlang в том числе) – кортежи. LXF

» **Через месяц** Рассмотрим применение кортежей в Erlang.

Админу сервера:

В первой части нашей новой серии уроков **Стюарт Бернс** демонстрирует, как сэкономить время и усилия, создав базовую стартовую установку.





Наш эксперт

Стюарт Бернс – британский администратор Linux в компании Fortune 500; его конек – сборка и поддержка крупномасштабных виртуализованных инфраструктур Linux.

Установить дистрибутив Linux с CD-ROM легко. Но если вы начнете устанавливать десятки и сотни систем, то уже сложнее гарантировать согласованность, и сборка всех машин вручную менее осуществима – она отнимет слишком долгое время. И потом, есть еще настройки, которые нужно внести. Чуточка планирования и автоматизации позволит вам быстро создавать однотипные установки снова и снова.

В данной серии уроков я надеюсь наметить основные сборочные блоки автоматизации быстрого развертывания [kickstart – *букв.* «вбрасывание пинком»], и вы сможете создавать системы по требованию за несколько простых шагов.

Чтобы наши упражнения пригодились в реальном мире, мы соберем систему автоматизированной установки, основанную на наборе систем для быстрого развертывания внутри виртуального окружения. Это также идеально подойдет для быстрой сборки серверов по требованию в вашем персональном окружении с малым ручным вмешательством.

Для экспериментов с автоматическим развертыванием, в любом значимом смысле, вам потребуется web-сервер – и для конфигурации, и для файлов установки. Мы возьмем CentOS 5, поскольку для наших целей он идентичен Red Hat, но бесплатен.

Давайте построим базовый web-сервер и заготовим базовую автоматизированную установку. Соберите/получите оборудование для web-сервера, виртуальное или нет, если вам нужно скачать его с http://www.centos.org. По возможности берите DVD-версию, так как она уложила все необходимое на единственном диске. DVD доступен в основном через скачивание посредством BitTorrent, так что воспользуйтесь клиентом BitTorrent.

Мы создадим конфигурацию минимального web-сервера и на нем будем основывать наш скрипт.

Вставьте ваш носитель с CentOS в привод CD-ROM (реальный или виртуальный) и запустите установку Centos, выбирая правильные опции вашего географического положения, уникальные пароли и прочее. Если вы используете скачанный диск впервые, стоит позволить серверу выполнить тест носителя и убедиться, что ваша копия – не брак.

В конце концов вы доберетесь до разметки диска. Советую выбрать «Remove all disk partitions on selected drives and create a default layout [Удалить все разделы на выбранных дисках и создать разбиение по умолчанию]». При нажатии Next вас предупредят, что все данные будут стерты. Если вы согласны, жмите OK.

Когда дело дойдет до разметки дисков, в нашей демонстрации установите размер диска на 10–15 ГБ (полагаем, что все папки находятся на одном разделе), так как потом мы скопируем все содержимое установочного диска CentOS на web-сервер, расположенный в **/var**.

Единственное изменение, которое следует сделать – использовать статический IP-адрес для нашего web-сервера. Запишите используемый IP-адрес; позднее он пригодится.

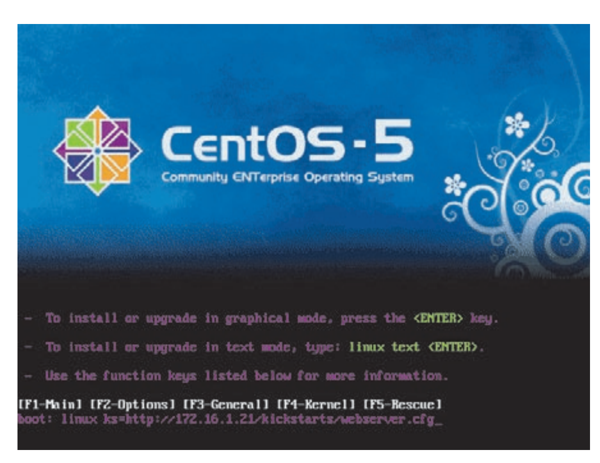
На стадии выбора пакетов я предлагаю взять те, что выписаны ниже. Чтобы выбрать отдельные пакеты, выберите переключатель «Customise now», и это даст вам больше контроля над пакетами.

» **Рабочий стол** (Gnome)

» **Сервер** Web-сервер

» **Приложения** (только Редакторы)

Когда при установке настанет черед брандмауэра, убедитесь, что http разрешен через порт 80. Наконец, понадобится



› **По сути, мы просто велим компьютеру использовать готовый запуск, без интерактивности.**

» **Месяц назад** Мы научились создавать систему аварийного восстановления.

Быстрый запуск

установить автозапуск **httpd** (web-сервер *Apache*), что сэкономит несколько минут при поиске неполадок на линии. Проделайте это, вызвав следующую команду в терминале от имени root – вызвать нужно только один раз, по завершении установки и перезагрузки сервера:

```
chkconfig --level 35 httpd on
```

Вам также потребуется запустить службу для первого раза, так как она пока не включена. Введите

```
service httpd start
```

Выполнив этот шаг, скопируйте содержимое установочного DVD в корневую папку web-сервера (**/var/www/html/**), поскольку мы собираемся использовать его для установки на базе web. Это можно проделать либо через GUI, либо по ssh, на ваш вкус. Для проверки, что все работает, просто возьмите другой компьютер и перейдите по адресу web-сервера. Вы должны увидеть каталог, где отображено содержимое, включая файлы с носителя, которые мы только что скопировали.

А теперь пойдут хитрости. При установке системы на базе Red Hat, Anaconda, инсталлятор Red Hat, сохраняет копию конфигурационного файла создаваемой установки в файл **/root/ks-anaconda.cfg**. И этим файлом можно воспользоваться для создания идентичной системы.

Он также пригодится для чернового восстановления системы до заведомо рабочего состояния, с последующей подстройкой из резервных копий. Однако нам требуется не просто клон системы: IP-адрес и имя хоста должны быть другие. Давайте приготовим нашу новую конфигурацию.

> «Для ускорения установки используйте текстовый режим.»

По требованию

Создайте папку в корне web-сервера (**/var/www/html**) под названием **kickstarts**. Здесь мы будем хранить варианты конфигураций, чтобы по запросу выбирать какой-либо из них для быстрого создания машин.

Скопируйте **/root/anaconda-ks.cfg** в свежесозданную папку **kickstart**. Поменяйте его название на более подходящее – например, **base.cfg**: ведь наши установки будут базироваться на этой главной конфигурации.

Большим плюсом этого способа является возможность припасти несколько различных конфигураций – скажем, web-сервер, dns-сервер и сервер баз данных. После этого уже просто останется загрузиться с CD и применить желаемый файл **kickstart config**, если экземпляр этой конфигурации уже создан.

Сделайте копию нового **base.cfg**, смените ее имя на **webserver.cfg** или любое другое имя, более осмысленное для вас – только без пробелов, ведь они только усложняют дело.

Задайте файлу такие права доступа, чтобы его мог читать web-сервер, командой

```
chmod 744 webserver.cfg
```

В нашем файле **webserver.cfg** нужно подправить некоторые параметры, чтобы мы могли выполнить web-установку. Откройте файл в вашем любимом редакторе, там мы поменяем установочный метод и значение IP.

По умолчанию все строки про разбивку диска Anaconda превращает в комментарии, в целях безопасности. Следовательно, нам нужно раскомментировать группы disk и volume и создание логического тома. Строки-комментарии начинаются со знака **#**.

Вы увидите строки, которые нужно раскомментировать (они начинаются с **clearpart**, а заканчиваются на **last logvol line**). Предупреждаем, что при такой конфигурации все имеющиеся диски будут затерты.

Также нужно менять IP-адрес новых серверов, поскольку нам ни к чему дублирование IP-адресов в нашей сети. Для этого обяжите конфигурацию получить во время установки DHCP-адрес. Найдите строки, начало которых похоже на следующее (ваш IP-адрес, и прочее, может отличаться, но вид строки вы узнаете):

```
network --device eth0 --bootproto static --ip 172.16.1.21
--netmask 255.255.255.0 --gateway 172.16.1.1 --nameserver
4.4.4.4,8.8.8.8
```

Замените это на

```
network --device eth0 --bootproto dhcp
```

Это гарантирует, что каждый новый web-сервер при развертывании получит новый, хотя и временный, адрес. Понятно, что если это производственный сервер, то после установки адрес следует поменять обратно на статичный.

Изменение метода установки делается вполне прямолинейно. Вторая строка файла, где прописано **cdrom**, и есть метод установки. Поскольку мы выполняем web-установку, сообщим об этом стартовой конфигурации, задав ей метод и url:

```
url --url http://url to web root/
```

Теперь методом установки будет **http** (web-установка).

Протестируем наш способ установки, собрав еще один сервер. Вставьте носитель с Red Hat в сервер, который мы хотим собрать, и загрузитесь. Попав в начальный текстовый экран выбора, измените на загрузку с файла kickstart. Это делается простым вводом команды (при необходимости замените имена файлов и папок):

```
linux ks=http://address of web server/kickstart/webserver.cfg
```

Сейчас система должна автоматически запускать установку и создавать новую систему с той же конфигурацией и тем же »

Устранение неполадок

Если ваше быстрое развертывание не сработало, попробуйте следующее – это подскажет, где искать:

» Все ли кабели на месте, идет ли пинг на web-сервер?

» На месте ли *Apache*, работает ли он (можно ли попасть в него)?

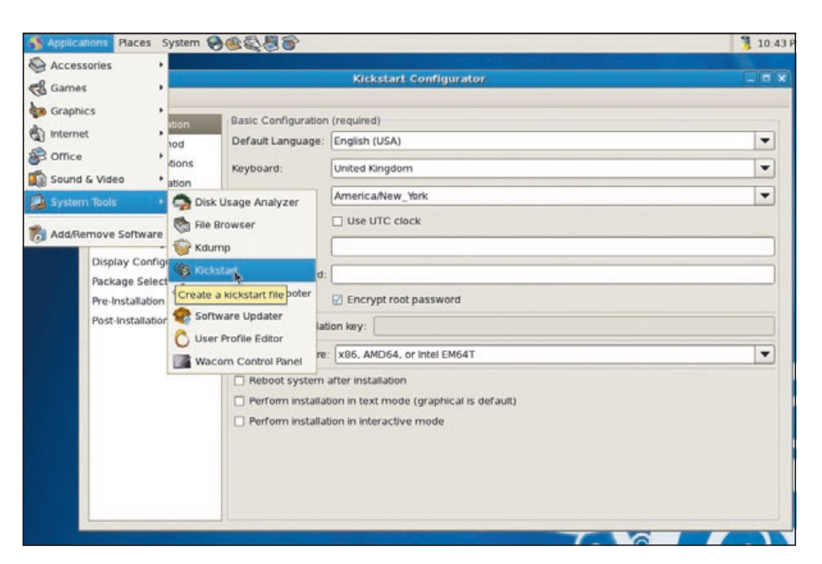
» Можете ли вы перейти и просмотреть kickstart, щелкнув по нему на web-сервере?

» Изменили ли вы kickstart, чтобы гарантировать его загрузку и использование DHCP в качестве адреса?

» Могут ли другие машины получить DHCP-адреса?

Скорая помощь

HTTP – не единственный способ установки. Можно также использовать NFS и FTP. При желании можно даже собрать индивидуальный DVD. Для их использования поменяйте параметр **--url** на NFS или FTP.



› Поменяйте экран настройки, чтобы загружаться по kickstart.

паролем; точная копия, если не считать IP-адреса. Поздравляем с первым успешным быстрым развертыванием; но давайте сделаем его интереснее.

Машины, собранные с такое конфигурацией, должны загружаться и получать DHCP-адрес. Хорошим началом для более продвинутой конфигурации будет сборка системы с нужной нам спецификацией. Итак, базовое развертывание работает – давайте исследуем конфигурацию kickstart. Ее можно разделить на четыре основных раздела:

» Предустановочный раздел (%pre)

Скрипты и команды, которые выполняются перед началом установки.

» Установочный раздел (install или upgrade)

Пункты, определяющие настройку системы, включая поясное время, пароли и конфигурацию дисков. Пароль может быть представлен в незашифрованном виде, если убрать **--isencrypted**, а после пароль открытым текстом. Это следует делать, конечно, только на время установки, а после установки сразу же менять пароль.

» Выбор пакетов (%packages)

Здесь вы определяете, какие пакеты будут устанавливаться. Символ **@** означает группы пакетов в том же стиле, что и группы пакетов rpm, а не отдельные пакеты. Отдельные строки означают отдельные пакеты, например, **yum install rpmfilename**.

Смачные команды

Безусловно стоит взять на вооружение **yum**: *Yum* предоставляет опции, упрощающие установку программ, и заботится о разрешении проблем в пакетах. Если вы собрались позабавиться пост-установкой RPM'ов, вот краткий перечень опций, способных вам пригодиться:

» yum localinstall localfile Устанавливает файл с локального диска (необходимо задать путь и имя файла).

» yum install rpmfile Устанавливает файл **rpmfile** из репозитория.

» yum search "string" Возвращает все пакеты, соответствующие строке string

» yum remove packagename Удаляет пакет. Применяйте с осторожностью.

Еще один полезный совет: имеет смысл принять во внимание, что по умолчанию ключ-подпись в базу данных не импортируется. Из-за этого при каждом запуске команд, связанных с **yum** и **rpm**, вас будут постоянно спрашивать, доверяете ли вы источнику. По умолчанию публичный ключ хранится в **/etc/pki/rpm-gpg**. Если его импортировать в базу данных, вопрос задаваться перестанет, поскольку ключ при этом добавляется в список доверенных ключей.

Вы не можете просто скопировать пакеты в одну папку **repo** на сервере. При желании установить дополнительный нестандартный пакет проделайте это в разделе пост-установки. Так вы предотвратите засорение папки установщика и необходимость заново создавать метаданные, содержащие манифест пакетов.

Настройка пакетов

Отредактировав файл пакетов, можно настроить, чтобы пакеты устанавливались по время стандартной установки системы. Знак **@** означает группу установки – например, группу пакетов по типу определяемой в **yum install system**. Отдельные пакеты можно установить путем простого добавления списка пакетов с новой строки.

» Пост-установка (%post)

Пользуйтесь этим для настройки сервера путем установки и запуска скриптов, пакетов и установок. Например, мы установили запуск http при загрузке вручную, но это можно сделать автоматически, прописав в разделе **%post**

```
chkconfig --level 35 httpd on
```

Это гарантирует, что web-сервер в этой конфигурации будет запускаться автоматически.

Вот еще один несложный пример:

```
yum -y install updates
```

Все локальные файлы обновятся с использованием автоматически выбираемого репозитория, который поставляется как часть Centos. Выполнение этой команды в качестве пост-установки потребует времени, так как обновления могут занимать более 300 МБ после первичной установки с носителя.

Другие полезные вещи, которые можно проделать, включают добавление в систему пакетов (например, если вы хотите поставить на новой машине утилиты для гостевой виртуализации). Следовательно, в этом небольшом сценарии мы установим добавления Guest (они же – Guest virtualisation tools), к нашему набору виртуальных клиентов, базирующихся на VirtualBox.

Создайте папку в корневом разделе web-сервера и назовите ее **postinstall**. В папке **postintall** web-сервера создайте папки с именами **packages** и **script**. Очевидно, в первой будут содержаться бинарники пакетов, а во второй – скрипты установки для этих пакетов. Скрипты сделаем ясными и простыми – легче будет отлавливать ошибки.

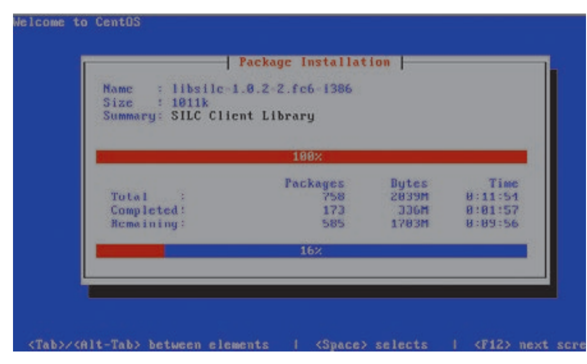
Как правило, установка гостевых утилит для VirtualBox – это скрипт, который должен запуститься при монтировании его через виртуальный CD-ROM в вашем виртуальном госте (при этом лучше получается настройка установки, выделение, копирование графики, и т. д.). Можно проделать это для всех гостей, но тогда пострадает наша идея автоматизации установки. К счастью, делается это очень просто. В следующем фрагменте я буду полагать, что вся ваша установка основана на Virtualbox.

Войдите в GUI web-сервера как root (есть и другие способы, но для наших целей этот самый быстрый). В меню Virtualbox выберите Devices > Guest Additions. При этом примонтируется CD, содержащий данные, которые нужно скопировать содержимое CD-ROM (или просто **VboxLinuxAdditions.run**) в папку **postinstall/packages**.

Чтобы установить гостевые службы на нашем сервере, необходимо также скопировать по сети наши файлы, сделать файл установки исполняемым и установить заголовки пакетов **gcc** и **kernel**. Эти пакеты – условие успешной установки гостевого сервера. Все это можно завернуть в один душевный скрипт, который сделает за нас всю работу.

```
yum -nogpgcheck -y install gcc kernel-devel.i686
wget http://url to server/postinstall/packages/VBoxLinuxAdditions.run /tmp/
```

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.



› Применяйте Kickstart Generator для создания или модификации kickstarts — меньше будет опечаток.

Наводим мосты

В виртуальных окружениях вроде VirtualBox вам может понадобиться поменять тип сети с NAT на сеть с мостами, чтобы разрешить доступ к интернету и LAN.

Как следует из названия, сетевой мост работает как связь между двумя сетями и разрешает трафику течь между ними.

Без сетевого моста у вас могут возникнуть проблемы с сетевым соединением.

Чтобы проделать это в VirtualBox, перейдите к требуемому клиенту, а затем в Settings. Оттуда выберите Network, после чего Bridged, и затем укажите используемый сетевой интерфейс (обычно eth0).

При желании можно использовать в вашей сети NAT (что и задано по умолчанию), которая создаст новую внутреннюю сеть на новой виртуальной локальной сети.

Какой бы способ вы ни применяли, убедитесь, что главный web-сервер и дополнительные клиенты расположены в одном сегменте сети и имеют одинаковый тип сети.

```
chmod 700 VboxLinuxAdditions.run
/VBoxLinuxAdditions.run
```

При запуске этот скрипт устанавливает в систему пакеты **gcc** (компилятор C) и пакеты разработчика ядра, чтобы собрать требуемые модули.

Ключ **--nogpgcheck** нужен, так как он велит *yum* не проверять цифровую подпись пакетов rpm. Пользуйтесь этим, только когда вы знаете, откуда взяты устанавливаемые пакеты, и осознавайте риск от игнорирования проверки цифровой подписи.

Опция **-y** довольно проста. Она велит системе установки *yum* считать, что на все запросы отвечают yes. Следовательно, она не станет спрашивать, хотите ли вы установить пакет, а просто устанавливает его. Без этой опции скрипту нужно будет интерактивное вмешательство.

Вторая строка просто копирует по сети файл, который мы хотим установить, и помещает его в каталог **/tmp**. В третьей строке устанавливается флажок запуска для владельца файла, что делает файл исполняемым. Затем, в четвертой строке, происходит сама установка гостевого ПО. Возможно, понадобится запустить это несколько раз вручную и убедиться, что все работает как полагается, а уж потом вписывать его в раздел **%post**. Раздел post не отображает раздел **%post** во время выполнения.

Сохраните этот скрипт в папке **/postinstall/scripts** под подходящим именем – я выбрал **guest.sh**. Этот пакет, при запуске его с только что собранного сервера, установит добавления для гостевой машины. Установка приложения выполняется в три этапа. Во-первых, скопируйте по сети скрипт с сервера, во-вторых, сделайте его исполняемым, и, наконец, запустите его. Чтобы проделать это нашей замечательной пост-установкой, сделаем небольшой скрипт из трех строк:

```
wget http://url to web server/postinstall/scripts/guest.sh /tmp/
chmod +x guest.sh
./guest.sh
```

Как и предыдущий скрипт, он копирует по сети, делает исполняемым и запускает его, но мы заодно используем его как пример, что можно сделать. В принципе, мы могли бы поместить его прямо в скрипт **%post**; ну, а если вы понапишете скрипты из сотен строк – что делать тогда? Этот подход также предотвращает установку ненужных пакетов, и размещение в скриптах меньших размеров помогает содержать все в порядке.

> **«Итак, базовое развертывание работает — исследуем kickstart.»**

В основном для этого упорядочивания мы создали папку **/postinstall**, а в ней папку **scripts**. Затем, используя стандартную утилиту **wget**, мы копируем файл **virtclient.sh** с сервера на только что созданную машину. После этого можно отмечать его как устанавливаемый пользователем root.

Теперь у нас есть копия приложения **install** – скопируем ее по сети в папку **postinstall**. Базовая установка готова к работе, и можно подумать о настройках.

Полезные вещи, которые можно проделать, включают добавление собственного репозитория на компьютер и выключение тех, что использовались в установке по умолчанию. Если у вас большое окружение и вы можете только применять изменения, это очень полезно. Присоедините это к скрипту пост-установки, и ваша машина будет обновляться и перезагружаться по мере того, как базовая конфигурация скопирована по сети.

Мы разобрались, из чего складывается быстрое развертывание; но есть и более эффективный способ изменять его файлы. Red Hat предоставляет GUI-программу, которая выполняет всю грязную работу через удобный интерфейс. Если вы любитель экспериментов, есть прекрасная программа, позволяющая создавать или обновлять **kickstart config**. Либо выберите ее из установки GUI-приложений, либо выполните от имени root в терминале следующую команду:

```
yum install system-config-kickstart
```

Я советовал бы установить эту программу на такой компьютер, где можно легко (посредством ssh или другими способами) копировать полученный kickstart в папку **kickstart** нашего web-сервера.

С помощью этого приложения можно добавлять и удалять пакеты, изменять такие критерии установки, как диски, группы томов, поясное время и любые другие опции в конфигурации kickstart, которые вы хотели бы настроить. **LXF**

В других дистрибутивах

Средства автоматизированной установки системы, аналогичные *kikstart*, есть и в других популярных дистрибутивах. Все они работают одинаково. Необходимо каким-либо образом создать файл конфигурации и при загрузке установочной системы сослаться на него.

В openSUSE такой механизм называется *AutoYaST*. Он позволяет сохранить описание установленной системы для дальнейшего клонирования в формате xml или создать и редактировать такой файл с помощью одного из модулей *YaST*. При загрузке установочной системы достаточно в параметрах добавить **autoyast=<URL файла xml>**.

В Ubuntu программа установки может брать необходимые параметры из текстового preceed-файла. Более подробно об этом можно прочитать здесь: https://help.ubuntu.com/10.10/installation-guide/i386/appendix-preseed.html

» Через месяц Настройка автоматического обновления и пост-конфигурации.





Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru

В этом месяце мы ответим на вопросы про...

1 Проблема Ntop

В Я загрузил *Ntop* и задал пароль. Программу в браузере я вижу, но попытки получить доступ к разделу администратора блокируются — мои имя пользователя и пароль не работают. Как мне их сбросить?

Мик Браун [Mick Brown]

О Пароль можно сбросить, удалив содержащий его файл, заново запустив исходную команду и задав новый. Имя файла – **ntop_pw.db**, и обычно он находится в **/var/lib/ntop**; если его там нет, обнаружьте его командой **locate**. Перед удалением файла выясните, кто его владелец. Затем удалите файл и запустите *Ntop* из терминала, от имени root, с такими опциями:

```
/usr/bin/ntop -P /var/lib/ntop -u ntop
--setadmin-password
```

Здесь подразумевается, что файл пароля находился в **/var/lib/ntop** и принадлежал пользователю **ntop** – если Ваша система настроена иначе, внесите должные изменения. Вам предложат задать пароль; сделайте это и закройте сеанс. Альтернатива – указать пароль в командной строке (тогда Вы будете видеть, что вводите, но будьте осторожны, потому что все те, кто находится рядом с Вами, тоже это увидят). Команды будут такие:

```
/usr/bin/ntop -P /var/lib/ntop -u ntop
--set-adminpassword=mynewpassword
```

Теперь перезапустите *Ntop* как обычно, хотя бы через диспетчер сервисов Вашего дистрибутива, и постарайтесь больше не забывать пароль. **НБ**



› *Ntop* предоставляет массу информации о загрузке сети, но помните свой пароль администратора!

2 Screen наготове

В Я захожу с работы на свой домашний компьютер Fedora 13 через SSH. Если надо запустить нечто продолжительное, я открываю сеанс *Screen*, запускаю задание и отключаюсь от экрана, а возобновляю сеанс уже дома.

Как-то я начал выполнять давно просроченное обновление *Yum* в терминале через SSH, и забыл предварительно запустить *Screen*. Мне нужно было уходить через 10 минут, и я думал, что все закончится быстро, но казалось, что оно вообще не кончается! Не то что это было катастрофой, но мой коллега покрикивал, чтобы я торопился.

Нет ли способа передачи активного процесса в сеанс *Screen* уже после его запуска? В идеале, я бы переключил задачу *Yum* на *Screen*, отключился, вышел из системы и уехал домой. Возможно ли такое?

Марк [Mark]

О Помимо очевидного варианта – извлечь из этого случая урок и больше не забывать загодя запускать *Screen*, есть еще пара возможностей. При закрытии оболочки все запущенные ею процессы закрываются тоже – так работает управление процессами в Linux. Существуют способы предотвратить это – например, *Screen*, а также запуск команды с опцией **nohup**, но в Вашем случае применять их было уже поздно.

В оболочке есть встроенная команда **disown**, отсоединяющая процесс от запустившей его оболочки. При работающей программе нажмите Ctrl+Z, чтобы приостановить ее, затем дайте команду **bg**, для перезапуска в фоновом режиме. Это полезно и само по себе, для случаев, когда Вы запускаете программу и забываете добавить **&** к командной строке. После этого дайте команду

```
jobs
```

Она выведет список задач, работающих в Вашей оболочке. Если Вы не запускали никаких других

Наши эксперты

» **Мы найдем ответы на любой вопрос — от проблем с установкой системы или модемом до сетевого администрирования; главное — спросить!**



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа **LXF** – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон
Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Грэм Моррисон
Когда Грэм не обозревает кучи программного обеспечения и не халтурит с *MythTV*, он готов к ответам насчет любого оборудования и проблем виртуализации.



Валентин Синицын
В редкие свободные минуты Валентин обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.



Юлия Дронова
Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования www.unixforum.org.

КУДА ПОСЫЛАТЬ ВОПРОСЫ

Пишите нам по адресу answers@linuxformat.ru **или спрашивайте на форуме:** www.linuxformat.ru

заданий в фоновом режиме, такая задача, вероятно, будет только одна. В начале строки идет номер задачи, и Ваша задача, скорее всего, будет иметь номер 1. Теперь ее можно отсоединить от оболочки следующей командой:

```
disown %1
```

Не забывайте предварять номер задачи знаком **%**. Это отсоединяет задачу от оболочки, которую Вы после этого сможете закрыть, не закрывая программу. Основной недостаток данного подхода – потеря вывода программы. Определить, работает ли программа, можно командой **ps**, но когда она завершится, не будет способа определить успешность ее выполнения. Альтернатива – применение программы *Reptyr*, доступной на сайте http://tinyurl.com/4tsfquc. Придется загрузить и скомпилировать исходный код (вряд ли какие-то дистрибутивы уже ввели *Reptyr* в свой состав). Получите идентификатор процесса (PID) программы с помощью *pgrep* или *pidof*:

```
pgrep programname
pidof programname
```

Переключитесь на другую оболочку или сеанс *Screen* и скомандуйте

```
reptyr PID
```

Программа переключится со старой оболочки на новую. Для большей уверенности, перед закрытием старой оболочки можно скомандовать **disown**, чтобы программа отсоединилась от исходной оболочки. В Ubuntu Maverick или более новой версии *Reptyr* откажется работать, из-за настроек безопасности, принятых по умолчанию. О преодолении этой проблемы см. на man-странице *Reptyr*.

Еще один подход – перед установлением SSH-соединения с Вашим компьютером позаботиться о запуске *Screen*. Это не просто ехидное замечание: можно добавить в свой профиль пару строк, запускающих *Screen* перед каждым запуском оболочки через SSH:

```
if [[ -n "${SSH_TTY}" ]] && [[ -z "${STY}" ]]; then
  screen -xRR
fi
```

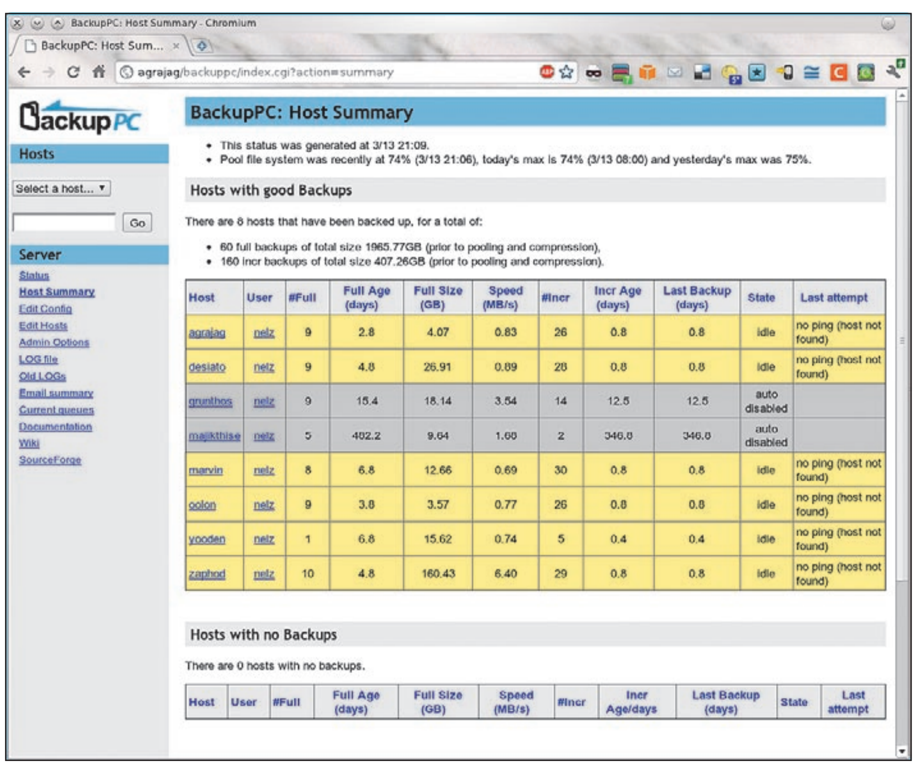
Теперь при каждом открытии сеанса SSH будет возобновляться существующий сеанс *Screen*, если таковой имеется, а в противном случае – создаваться новый. **ГМ**

3 Саботаж BackupPC

В Недавно я последовал вашему совету и установил *BackupPC* — на LinkStation, где обновил BIOS, чтобы установить Debian Etch — и настроил его на выполнение резервного копирования трех компьютеров Windows, принадлежащих моим жене и детям, а также моего собственного, с Ubuntu.

Резервное копирование компьютеров Windows выполняется надежно, но мне так и не удалось заставить *BackupPC* заработать с Ubuntu. Полагаю, проблема тут в SSH, потому что я получаю ошибку 'ListFilesReceive'. Я настроил SSH на доступ ноутбука с Ubuntu к LinkStation без пароля, и задал также обратное правило, но ничего не работает. Пока я обхожусь *Duplicity*, но хотел бы, чтобы всю работу выполнял сервер, а не мой ноутбук.

Ричард Мур [Richard Moore]



› **Для управления резервным копированием с *BackupPC* установите права доступа SSH на сервере и клиентах.**

О *BackupPC* работает от имени пользователя backuppc, но нуждается в доступе с правами суперпользователя-root к компьютерам, резервные копии которых требуется создать. Вам надо сгенерировать ключ SSH для пользователя backuppc на сервере и добавить его к файлу **authorized_keys** для пользователя root на каждом клиенте. На системе Ubuntu, где пользователь root обычно блокирован, это необычно. Но root там все-таки есть – у него просто нет пароля для входа в систему. Для *BackupPC* это не проблема, потому что здесь применяются не пароли, а ключи SSH.

Чтобы сгенерировать ключ на сервере, нельзя просто зарегистрироваться как пользователь backuppc: этот пользователь не имеет возможности входа в систему. Поэтому запустите с правами root команду

```
su -s /bin/sh -c "ssh-keygen -t rsa" backuppc
```

При желании в качестве типа ключа можно взять **dsa**, а не **rsa**; а если и на клиенте, и на сервере работает новейшая версия *OpenSSH*, можно использовать типы ключей **ecdsa**. Команда создаст в домашнем каталоге пользователя backuppc, обычно – **/var/lib/BackupPC**, два файла: **.ssh/id_rsa** и **.ssh/id_rsa.pub**. Второй файл, общий ключ, нужно добавить в список авторизованных ключей пользователя root на клиенте, поэтому скопируйте его на этот компьютер и дайте команды

```
sudo mkdir -p /root/.ssh
sudo cat id_rsa.pub | tee -a /root/.ssh/authorized_keys
```

Здесь нужно использовать **tee**, потому что стандартные методы перенаправления не работают с **sudo**, а перенаправление вывода команды **sudo** необходимо, поскольку команды запускаются не от имени root. *BackupPC* по-прежнему не сможет установить соединение, потому что при первой попытке соединения SSH с хостом тот запросит у Вас подтверждения его сигнатуры SSH, поэтому вернитесь на сервер и скомандуйте

```
su -s /bin/sh -c "ssh root@laptop" backuppc
```

По запросу введите **yes**. Эта операция выполняется только один раз – сигнатура сохраняется в **.ssh/known_hosts**, и теперь автоматизированное резервное копирование должно заработать. Если проблемы не прекратятся, попробуйте включить в конфигурационном файле настройку **$Conf{XferLogLevel}**. Чем выше ее значение, тем больше информации заносится в журнал. Чтобы журнал не заполнялся слишком быстро, советуем добавить эту команду в файл настройки конкретного компьютера в **/etc/BackupPC/pc**, чтобы лишняя информация не засоряла журналы компьютеров, работающих без проблем. **МС**

4 Мне не остановиться

В У меня возникла проблема, хоть и не критичная, но досадная. Почему-то мне никак не выключить компьютер, если только я не перехожу в Windows и не запускаю команду останова оттуда. Все операции и команды, которые мне советовали попробовать, выводят меня в меню двойной загрузки. Мой компьютер — HP Pavilion Elite HP410f, на нем стоят Ubuntu 10.10 и Windows 7.

Харли Диллз [Harlie Dills]

О Начнем с того, что для выключения компьютера не обязательно загружать Windows. Перезагрузка тоже корректно завершает работу ОС, вот что для Вас важно. В ин- »

тервале между перезапуском и появлением загрузочного меню можно нажать кнопку питания и выключить компьютер. Это рискованно только тогда, когда Вы уже выбрали один из пунктов загрузочного меню (или один из них был выбран по умолчанию по причине тайм-аута).

Теперь поищем правильное решение. Проблема, скорее всего, с интерфейсом ACPI (Advanced Configuration and Power Interface), и кроется в конфликте между системой ACPI Linux и BIOS Вашей материнской платы. Первым делом зайдите на сайт изготовителя материнской платы и гляньте, нет ли там обновлений BIOS – обычно они есть, потому что платы редко поставляются с новейшей версией BIOS. Обновления BIOS могут исправить все виды странных проблем, поэтому стоит иметь новейшую версию.

Если это не поможет, следующим шагом попытайтесь временно блокировать всю поддержку ACPI в ядре. В меню загрузки выделите пункт, соответствующий Ubuntu, и нажмите клавишу **e**, чтобы начать редактирование. Найдите строку, начинающуюся с подстроки **linux**, и припишите в ее конец подстроку **acpi=off**. Нажмите Ctrl+X, чтобы загрузиться с этим параметром. Если выключение сработает, это подтвердит наличие проблемы ACPI. Но совсем блокировать ACPI Вам вряд ли захочется. Попробуйте добавлять следующие опции в пункт загрузочного меню, поодиночке или в комбинации, пока не определите, какая из опций или их комбинаций является рабочей.

```
acpi=ht
pci=noacpi
acpi=noirq
pnpacpi=off
noapic
nolapic
```

Первая опция достаточно сурова: она блокирует львиную долю функций ACPI, так что не пользуйтесь ею без крайней необходимости.

Найдя решение, сделайте его опцией по умолчанию – не править же параметры ядра при каждой загрузке! Отредактируйте **/etc/default/grub** от имени root: найдите элемент для **GRUB_CMDLINE_LINUX** – вероятно, Вы распознаете содержимое как опции ядра, к которым Вы делаете добавку. Вставьте сюда Ваши работающие параметры ядра, сохраните файл и введите изменения в силу, скомандовав

```
sudo update-grub
```

PH

5 Обновление-чудовище

В Начиная работать с Linux, я установил Fedora Core 1 и, получив возможность установить все, я так и сделал. Провозившись с получением *Sendmail*, *Apache*, FTP, PHP, *MySQL* и остального, я бы хотел сохранить настройки и все прочие ассоциированные с ними файлы. Система их уже не обновляет, поскольку они чересчур устарели, а я хотел бы извлечь преимущества нововведений в платформе. Требуемые изменения – например, обновления PHP с 4.x на 5.x, *MySQL*, улучшенная система безопасности, исправления в ядре, и т.д.

Недавно у меня появился свободный компьютер, и я загрузил Fedora Core 5, затем обновился до Core 6 или 7 и пришел в ужас: обновление отняло 10 часов. Мой сервер в общем не для критичных задач, но я не хочу парализовать его на целый день. Можно ли рассчитывать, что при обновлении сохранятся все скрипты и файлы настройки? Можно ли обновиться с Fedora Core 1 на Fedora 14? Сохранятся ли файлы настройки для пакетов типа sendmail **и** httpd**? Сохранится ли все ПО, часть которого установлена в каталогах наподобие** sbin**, а не только в пользовательских каталогах?**
Тони Райдер [Tony Rider]

О Fedora Core 1 – дистрибутив явно доисторический, и обновление с него до новейшей версии – непростой процесс. Прямое обновление почти наверняка обречено на провал, а обновление через промежуточные версии, если Вы сумеете их найти, продлится целую вечность. *Glibc*, ключевая библиотека в системе Linux, претерпела немало изменений, и Вы замучаетесь последовательно обновлять ее через все промежуточные версии. Поскольку Вам нужен работающий компьютер, а не победа над техническими сложностями, единственно разумным подходом будет полное резервное копирование всех Ваших данных и файлов настройки с последующей установкой новейшего дистрибутива с нуля. Резервное копирование пользовательских данных пройдет очень просто; если у Вас отдельный раздел **home** – просто не переформатируйте его. В противном случае создайте резервную копию всего содержимого **/home** на внешний (или сетевой) диск и восстановите его после установки – при установке обязательно создайте отдельный каталог **/home**.

Вы упомянули *Sendmail* и *MySQL*; они хранят свои данные в каталоге **/var**, поэтому он тоже подлежит резервному копированию, а по завершении установки – восстановлению на отдельный раздел. Данный подход выглядит трудоемким, но альтернативный вариант отнимет еще больше времени и только вызовет разочарование. Вам нужно создать резервные копии содержимого **/etc** и **/var**, а **/sbin** проблем не доставит, потому что там только программы. Затем выполните новую установку и импортируйте все данные и настройки в новую систему. Если Ваши программы не настолько устарели, что с тех пор сменился способ хранения данных или настроек, то новое ПО должно позволить Вам импортировать старые файлы. Возможно, в процессе Вы получите различные сообщения об ошибках, но предложенный метод все же действительно самый простой.

Чтобы убедиться, что у Вас установлено то же самое ПО, что и ранее, сгенерируйте список установленных программ с помощью **rpm**:

```
rpm --query --all
```

Перенаправьте вывод этой команды в файл или на принтер – по нему Вы потом проверите наличие программ. Сравните содержимое файлов **/etc/passwd** в обеих системах и убедитесь, что идентификаторы пользователей (user ID) одинаковы. Если это не так, измените владельцев файлов, иначе будут проблемы с правами доступа. **НБ**

6 UPS и NUTs

В У меня есть UPS CyberPower, и документация, поставляемая в комплекте, заявляет о поддержке Linux, которая на деле не обеспечивается. Я установил пакет, поставляемый в комплекте, а когда выяснилось, что он не работает – то и бета-версию пакета.

Я вновь посетил сайт производителя, загрузил с него Deb-пакет и попытался установить его, но получил сбой (в *KpackageKit*), со следующим сообщением:

```
An error occurred that we weren't expecting
```

При запуске pwrstat **программа утверждает, что демон не запущен, но вывод команды** ps **показывает, что демон** pwrstatd **работает:**

```
jseidel@EDP15:~$ pwrstat -status
Daemon service is not found.
jseidel@EDP15:~$ pgrep pwr
root 1480 0.0 0.0 8740 792 ? S 14:16 0:00 /usr/sbin/pwrstatd
```

Я пытался поработать с поставщиком, но тип из техподдержки упорно «передает вопрос разработчикам», а те вообще не отвечают. Не могли бы вы предложить мне какой-нибудь вариант корректного останова на случай сбоя питания?
Йон Зайдель [Jon Seidel]

О Вот что бывает, когда компании не понимают Linux и Open Source. Они пытаются писать собственный драйвер, вероятно, взяв за основу ПО для Windows, поскольку так все работает в мире Windows. (Мы воздержимся от комментариев, хорошо ли сообщение об ошибке отражает этот факт.) Они тестируют свое ПО с одной-двумя версиями пары дистрибутивов и считают работу сделанной; а если в Вашем дистрибутиве все устроено по-другому, то Вам не повезло. Пора им понять, что давным-давно существуют проекты Open Source, делающие то, что им нужно, и для поддержки их продукции достаточно передать необходимую информацию соответствующим командам,.

В случае с ПО для UPS, Вам требуется пакет **Network UPS Tools** или **NUT** (www.networkupstools.org). Он поддерживает ряд моделей CyberPower, хотя получил информацию методом обратной разработки.

Кроме текущей поддержки, *NUT* позволяет работать с несколькими UPS, что может пригодиться, если Вы решите поставить оборудование от другого изготовителя. Установите *NUT* через менеджер пакетов Вашего дистрибутива, а затем отредактируйте файлы настройки в **/etc/nut**, чтобы они отвечали Вашей модели UPS. В файле **ups.conf** заведите раздел для Вашего устройства в следующем формате:

```
[<upsname>]
driver = <drivername>
port = <portname>
```

Имя для Вашей UPS сойдет любое, а драйвер зависит от конкретного устройства CyberPower – просмотрите раздел совместимости на сайте *NUT*. В качестве порта обычно используется **/dev/ttyS0** – первый последовательный порт. Добавьте в файл **upsd.users** пользовательскую запись – она опре-

делит входное имя и пароль, разрешающие выполнять административные действия – например, выключение компьютера:

```
[<username>]
password = <pass>
upsmon master
```

Затем свяжите все воедино, добавив в файл **upsmon.conf**:

```
MONITOR upsname@localhost username pass
master
```

В файле **nut.conf**, либо оставьте для **MODE** значение **none**, чтобы Ваш дистрибутив запускал серверы *upsdrv*, *upsd* и *upsmon*, или установите для **MODE** значение **standalone** и запустите общий сервис *NUT*. Запустите сервисы и отключите шнур питания от UPS, следя за файлами журналов, чтобы убедиться в том, что все работает. Затем изучите другие опции – например, отправку уведомлений о событиях. ГМ

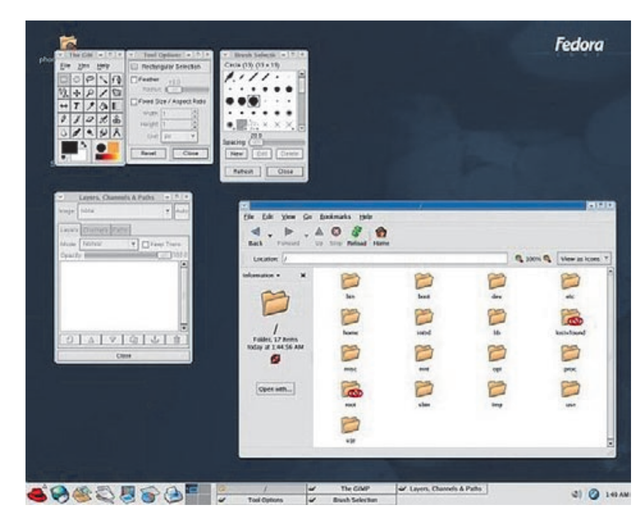
7 Нужен вентилятор

В На моем ноутбуке Acer 5315 была предустановлена XP. Я решил избавиться от Windows и установить Ubuntu 10.10. С установкой проблем не было, но потом процессор перегрелся, и через 10 минут система отключилась, поскольку в Ubuntu вентилятор немедля отказался работать. Поиск в Google обнаружил немало товарищей по несчастью, с той же проблемой — и с такой же моделью, как и у меня, и с другими.

Решать проблему предлагалось двумя методами: либо посредством редактирования *Grub 2*, либо обновлением BIOS. Обновление BIOS — шаг несколько рискованный, и некоторые люди, решившиеся на него, сообщили, что он не помог. Поэтому я решил перепробовать варианты редактирования *Grub*, но ни один не сработал. Похоже, Linux-программы для управления вентилятором нет, и в системе 10.10 тоже нет ничего подходящего. Мне приходится охлаждать ноутбук, подкладывая под него пузырь со льдом, чтобы работать подольше. Не присоветуете ли способа все-таки заставить вентилятор работать?

Терри Андерсон [Terry Anderson]

О Обновление BIOS отнюдь не настолько рискованно, как гласит распространенное мнение, особенно если речь идет о ноутбуках. Самый серьезный риск вызван возможностью сбоя питания в ходе обновления, а ноутбук, работающий от электросети, но с исправной батареей, защищен от этой случайности. Материнские платы редко выпускаются с BIOS, полностью свободной от ошибок. Частенько обновление оказывается доступным еще до того, как первая поставка продукции достигнет магазинов, поэтому обновление – это, в общем случае, дело хорошее.



› Не будем говорить, что Fedora Core 1 устарела, но ее обои по умолчанию не моложе птеродактиля. Не пытайтесь ее обновлять, если рассудок вам дорог.

Ваша проблема вызвана необычным решением Acer управлять вентилятором через разработанное ими Windows-приложение, а не аппаратно или через ACPI. Установка любой другой ОС, включая и обычную версию Windows, приводит к отказу вентилятора. Если обновление BIOS не поможет, попробуйте скрипт, который должен решить проблему. Загрузите его с сайта http://ubuntuforums.org/showpost.php?p=4925392, распакуйте архив и отредактируйте скрипт *acer_fancontrol*, подогнав его к Вашей системе. Комментарии в этом скрипте объясняют различные опции. »

Часто задаваемые вопросы

Fuse

» Это ведь не связано с защитой компьютера от сбоев питания? [fuse по-англ. в т. ч. бытовой электропредохранитель, «пробка», — *прим. ред.*]

Не связано. Fuse означает Filesystem in USErspace [файловая система в пространстве пользователя]. Это модуль ядра, позволяющий файловым системам запускаться в пространстве пользователя вместо пространства ядра.

» Ну и в чем тут фишка?

Традиционные файловые системы, типа ext3, XFS или ISO9660, входят в ядро Linux – они либо встроены в ядро, либо устанавливаются в виде модулей, включаемых в ядро при загрузке. Это нормально для файловых систем, но означает, что либо их код должен быть включен в главное дерево ядра, как с указанными тремя, либо нужно скомпилировать код специально для этого ядра. Обновите ядро, и все ваши внешние модули не будут работать, пока вы их не восстановите, как печально известно многим владельцам видеокарт NVidia.

» Чем Fuse от них отличается?

Fuse – тоже модуль ядра; он является частью официального ядра Linux с версии 2.6.14. Он обеспечивает мостик между пространством пользователя, где запускается файловая система Fuse, и самим ядром, позволяя использовать файловые системы, не являющиеся частью ядра.

» А мне какая с этого польза?

Да в общем небольшая, если только вы не решите разработать собственную файловую систему. Но это не значит, что другие не захотят такого сделать, и существует много файловых систем, которых без Fuse могло бы и не быть.

» Есть уже и Ext2/3, и ReiserFS, и XFS — зачем нам еще много файловых систем?

Если вы так ставите вопрос, то незачем. Но Fuse очень часто применяется для создания виртуальных файловых систем.

» Тут и с реальными-то системами не все ясно — а что такое виртуальные?

Виртуальные файловые системы не существуют на диске, но предоставляют информацию с помощью файлов-интерфейсов. Fuse предоставляет множество таких: *SSHFS* позволяет монтировать каталоги на удаленном компьютере, используя только SSH; *EncFS* дает зашифрованную файловую систему, и отдельные каталоги могут быть зашифрованы и смонтированы через *EncFS* для доступа. *NTFS-3G* обеспечивает полное чтение/запись NTFS-разделов Window – это одна из немногих файловых систем Fuse, являющаяся файловой системой в традиционном смысле этого слова.

» А больше Fuse ничего не умеет делать?

Существуют файловые системы для монтирования различных онлайн-сервисов, как если бы они были местными каталогами. Можно подмонтировать системы хранения онлайн или запасник фотографий, вроде Flickr. Если ваш цифровой фотоаппарат не монтируется как USB-накопитель, и вам необходим *Gphoto2* для извлечения фотографий, можете использовать файловую систему Fuse для монтирования.

» Где я могу узнать больше?

Домашняя страница Fuse – http://fuse.sourceforge.net, а список поддерживаемых файловых систем – на вики Fuse, http://tinyurl.com/94gza.

Чтобы убедиться в работоспособности скрипта, запустите его командой

```
sudo ./acer_fancontrol
```

Если все в порядке, поместите его и скрипт *mempat* в **/usr/local/bin** и отредактируйте **/etc/rc.d/local** с правами root:

```
sudo cp acer_fancontrol /usr/local/bin
sudo cp mempat /usr/local/bin
sudo gedit /etc/rc.local
```

Перед строкой **exit 0** добавьте следующую строку:

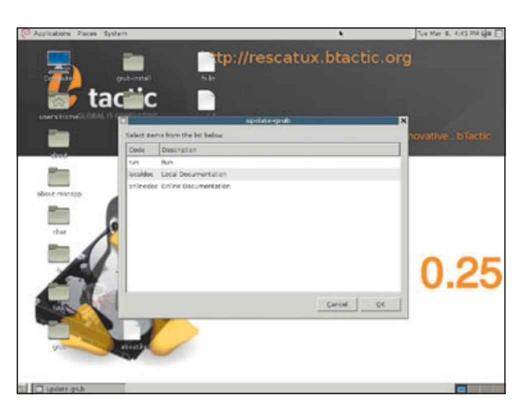
```
/usr/local/bin/acer_fancontrol
```

Благодаря этому скрипт управления вентилятором будет автоматически запускаться при загрузке. Хотя скрипт приведен на форумах, посвященных Ubuntu, ничто не мешает ему работать с любым другим дистрибутивом, но метод его запуска при загрузке для дистрибутивов, основанных не на Debian, будет отличаться от приведенного здесь. **ГМ**

8 Потеря Windows

Я решил попробовать Lubuntu с LXFDVD142, создав систему с двойной загрузкой, где второй системой была Windows XP. Обе системы работали правильно, и все было хорошо, пока менеджер обновлений не обновил ОС. После перезагрузки опция XP исчезла из меню загрузки – и теперь XP у меня нет.

Я попробовал исправить ситуацию, воспользовавшись процессом, который был описан в одном из ваших журналов [Grub 2: Польза Super Grub Disk, LXF128]. В двух строках имелись опечатки; я знаю, что вместо unmount **следует использовать** umount**, а вместо** mn – mnt**. Есть ли там другие опечатки, я сказать не берусь, но процесс, запущенный в терминале, других ошибок не зарегистрировал. Но опции XP в загрузочном меню все равно нет!**



› На помощь, *Super Grub Disk*! Держите эту программу под рукой на случай, если что-то (или кто-то) испортит ваше загрузочное меню.

Затем я переустановил Lubuntu и восстановил XP. И снова все было хорошо, пока не вмешался менеджер обновлений. XP опять исчезла. С помощью *Clonezilla* я восстановил XP с внешней резервной копии. Однако я хотел бы продолжать с Lubuntu, если вы поможете мне решить проблему.

Брайан Митчелл [Bryan Mitchell]

Приносим Вам наши извинения за опечатки в статье. XP Вы не потеряли – на самом деле потерялась только ее опция в загрузочном меню. Пошаговые инструкции в руководстве по *Grub 2* должны восстановить правильное меню загрузки – похоже, что сбой выдает модуль обнаружения ОС, *OS-prober*. *Grub* сам генерирует строки меню для копии Linux, скриптом из **/etc/grub.d/10-linux**, но другие ОС (например, Windows) обнаруживаются особой программой, *OS-prober*, которая вызывается другим скриптом из каталога **/etc/grub.d**. Проверьте, находит ли *OS-prober* установленную копию Windows, командой

```
sudo os-prober
```

Вывод команды должен выглядеть приблизительно так:

```
/dev/sdb1:Windows Vista (loader):Windows:chain
/dev/sdb2:Windows 7 (loader):Windows1:chain
```

где каждая строка соответствует обнаруженной копии Windows. Чтобы эта команда работала, разделы Windows не обязательно должны быть примонтированы. Эту информацию использует *Grub* для построения элементов загрузочного меню, и если программа *OS-prober* дает ошибку, то источник проблемы найден.

В отчетах об ошибках сообщалось о неработающем *Grub* и проблемах с *OS-prober* в некоторых конфигурациях Ubuntu 10.10. Один из методов устранения этих ошибок – настроить *Grub* программой *Super Gru Disk* (www.supergrubdisk.org). Загрузитесь с CD и в появившемся диалоговом окне выберите Grub Options. Вам на выбор предложат две опции – первая восстанавливает *Grub*, уничтоженный (пере)установкой Windows. Вторая опция, Update Grub Config, выполняет именно то, что требуется Вам. Выберите ее, и опция загрузки Windows в загрузочном меню восстановится. **НБ**

Терминалы и суперпользователи

Мы часто предлагаем решение в виде команд, вводимых в терминале. Хотя обычно то же самое можно выполнить и через графические инструменты, входящие в состав дистрибутива, отличия между этими инструментами означают, что такие решения будут узко специфичными. Команды терминала гораздо более гибкие и работают во всех дистрибутивах.

Команды настройки системы часто должны запускаться от имени суперпользователя (superuser или root). В зависимости от вашего дистрибутива, есть два способа сделать это. Многие дистрибутивы – в частности, Ubuntu и его производные – требуют предварять такие команды префиксом **sudo**. После этого пользователю предлагается ввести пароль, и система выдает ему полномочия root, действующие на время выполнения команды. Другие дистрибутивы используют префикс **su**, применение которого требует ввода пароля root и дает полные права root до тех пор, пока пользователь не введет команду **logout**. Если ваш дистрибутив использует **su**, выполните эту команду один раз, и потом сможете выполнять любую команду, не предваряя ее префиксом **sudo**.



Краткая справка про...

Компиляцию ПО

Большинство дистрибутивов имеют огромные репозитории готовых сборок почти всех требуемых вам программ. Но иногда бывает нужно установить пакет, скомпилировав его из исходного кода. Обычные причины для этого – отсутствие новейшего пакета для вашего дистрибутива или необходимость модификации исходного кода. Для этого сначала распакуйте исходный код из архива следующей командой:

```
tar xvf foo-1.2.3.tar.gz
```

Команда **tar** умеет определять, как сжат архив, и вам нет нужды сообщать ей об этом. Исходный код обычно распаковывается в каталог с тем же именем, что и архив. Перейдите туда командой **cd foo-1.2.3**. Отыщите файлы с именами **README** или **INSTALL** – обычно там есть инструкции по установке. Стандартная процедура выглядит так:

```
./configure
make
su make install
```

Первая команда проверяет систему, чтобы убедиться в наличии всех необходимых зависимостей и установить все опции программы. Для просмотра доступных опций неплохо предварительно запустить команду

```
./configure -help.
```

Вторая команда компилирует ПО, помещая создаваемые файлы в текущий каталог. Для настройки и компиляции не обязательно быть суперпользователем, но на третьем шаге выполняется копирование файлов в системные каталоги, а тут уже нужны права root. Поэтому для запуска третьей команды используется префикс **su**, благодаря которому команда выполняется с правами root. Пользователи Ubuntu вместо нее должны использовать команду:

```
sudo make install
```

Если команде **./configure** не дано других указаний, скомпилированная программа обычно устанавливается в **/usr/local/bin**. Если у вас RPM-система, и **configure** сообщает об отсутствующей библиотеке, но вам точно известно, что она есть, установите соответствующий пакет **-devel** – например, **libbar-devel**.

Большой вопрос Какой лучший метод избавиться от старых файлов?

В Когда появляется новый дистрибутив, я предпочитаю обновлять свою систему, сохраняя домашний каталог и все настройки, чтобы не создавать резервные копии и переконфигурировать все после установки. Я вполне доволен этой процедурой, но из-за нее у меня на диске скапливается множество старых файлов. Нет ли какого-нибудь (по возможности) простого и (очевидно) безопасного метода очистить систему от старого хлама?

Мартин Уэллс [Martin Wells]

О Эта задача имеет две подобласти: системные файлы и домашний каталог. Неиспользуемые пакеты склонны при удалении оставлять файлы, засоряющие **/etc**. В большинстве дистрибутивов есть специальный инструмент для их чистки. В дистрибутивах на основе Debian запустите *Synaptic* и щелкните по кнопке Status. Если у Вас уцелели старые файлы настройки, Вам будет доступна опция Not Installed (Residual Config); выберите ее и перечисленные пакеты, и *Synaptic* удалит файлы.

Еще один быстро заполняющийся каталог – **/var**. Файлы журналов в **/var/log** могут расти очень быстро, особенно если что-то не в порядке, поэтому обязательно установите *logrotate*. Эта программа архивирует и сжимает последние записанные файлы журналов и удаляет старые, препятствуя превращению **/var/log** в черную дыру. Можно также поискать в **/var/lib** старые файлы данных, оставленных серверами, которые Вы не используете. Если у Вас есть программы, использующие *MySQL* для хранения данных, может оказаться, что в **/var/lib/mysql** сидит множество файлов **mysqld-bin**. Для обычных пользователей настольных ПК хранение этих файлов – пустая трата места на диске. Отключите опцию их сохранения – отредактируйте **/etc/mysql/my.cnf**, закомментировав строку **log-bin**, и перезапустите *MySQL*. Старые файлы придется удалять, но новые создаваться уже не будут.

В каталоге **home** проще всего найти давно не использовавшиеся файлы командой **find**:

```
find ~ -mtime +365
```

Эта команда найдет все файлы, не модифицировавшиеся в течение года. Тем не менее старые файлы могут быть все еще нужны. Если ваш каталог **home** не монтировался с опцией **noatime**, то каждый файл содержит запись о том, когда его в последний раз открывали для чтения – это более полезно. Команда

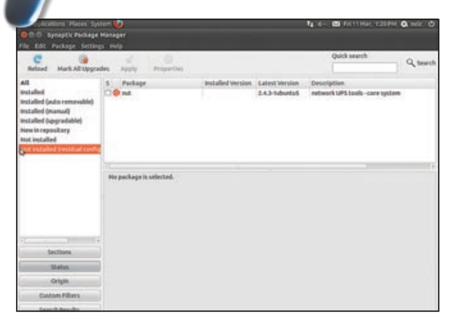
```
find ~ -atime +365
```

обнаружит все файлы, не открывавшиеся за последний год. Запуск команды **mount** в терминале покажет опции монтирования каждого файловой системы; если среди опций файловой системы, содержащей **/home**, нет **noatime**, воспользуйтесь этим методом. Для **find** имеются графические оболочки, например, *KFind*, находящаяся в самом низу главного меню KDE, но никогда ничего не удаляйте, не будучи абсолютно уверены в его ненужности. Если сомневаетесь, создайте резервные копии файлов перед их удалением:

```
tar czf abouttobegone.tar.gz $(find ~ -atime+365)
```

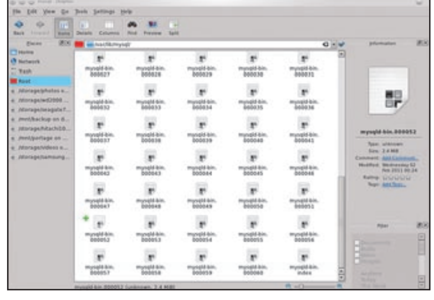
Ваша задача имеет и специальное решение – программу *BleachBit* (http://bleachbit.sourceforge.net), которая автоматически выполняет большую часть работы. Она не будет сканировать все файлы в вашей системе, как Вы с помощью **find**, но выявит наиболее вероятных кандидатов на удаление – например, временные файлы браузера. Для пущей безопасности *BleachBit* может истребить файлы так, что их будет не восстановить. Но это замедлит процедуру. **НБ** LXF

Шаг за шагом: Очистка от ненужных файлов



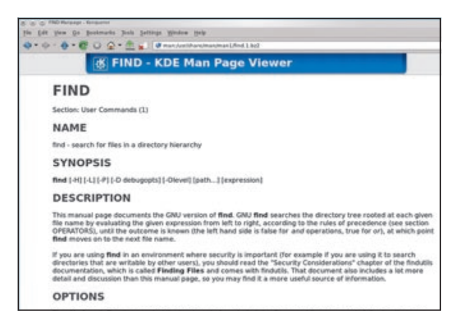
1 Удалите старые файлы настройки

Synaptic может удалить файлы настройки пакетов, которые вы удалили; это не только повысит безопасность, но и сэкономит дисковое пространство.



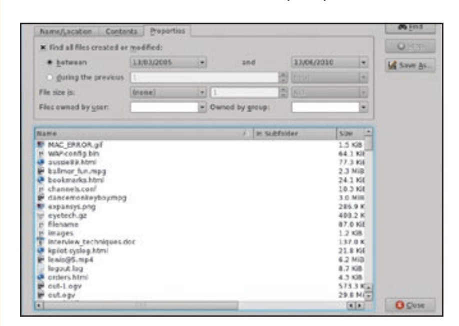
2 Двоичные журналы MySQL

Настройка **log-bin** в **my.cnf** может вызвать генерацию большого объема данных в **/var**, а там совсем не желательно позволять исчерпание дискового пространства.



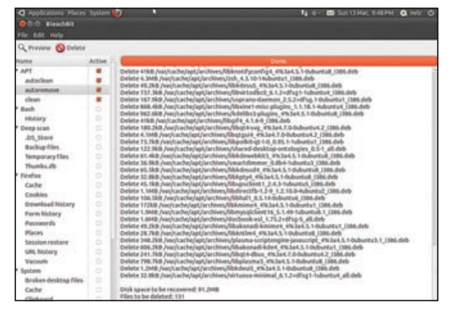
3 Найдите старые файлы

Команда **find** с опциями **-mtime** и **-atime** поможет найти все старые файлы, которые больше не нужны. Другие полезные опции указаны на man-странице.



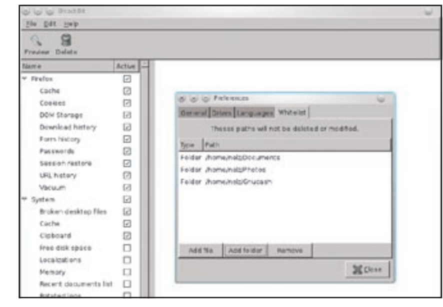
4 Можно и графически

Программы типа *KFind* предоставляют графический интерфейс для команды **find**. Пользоваться им проще, но вы не получите всех возможных опций.



5 Очистка мусора

BleachBit помогает найти файлы, созданные теми программами, что имеют обыкновение оставлять за собой хвост временных файлов.



6 Обойти ряд областей

Программе *BleachBit* можно дать указание игнорировать некоторые файлы или каталоги. То же самое можно сделать и с помощью опции **-prune** команды **find**.

LXFHotPicks



Ник Вейч
Компилируя HotPicks, Ник частенько употребляет *Curses*. Пошлите на известный адрес сообщение о своем любимом языке программирования.

Музыкальный плейер

Clementine

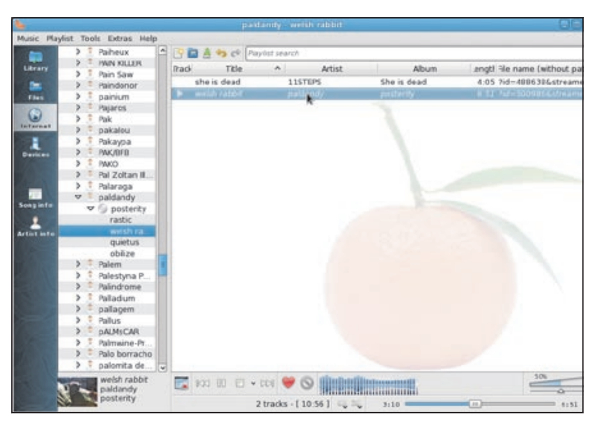
Версия 0.7 **Сайт** http://www.clementine-player.org/

Музыка... Она существует явно не для слушания, а для того, чтобы ее коллекционировали, упорядочивали, сводили в файлы, составляли рейтинги, перетасовывали и присваивали ей номера. По крайней мере, к такому выводу приходишь, глядя на объем (о Боже) приложений для воспроизведения музыки, засоряющих рабочие столы линуксолюбивой публики. Одни пришли и ушли, другим стоило уйти давным-давно, но всегда остается место для новых попыток.

В свете всего этого, *Clementine*, похоже, не слишком далекое подобие *Rhythmbox*, или *Banshee*, или *Amarok*... хм... минуточку. Ну да, он не очень отличается от *Amarok* – не нового *Amarok* пост-KDE4, а от старого, примерно версии 1.4. Те, у кого глаз орлиный, а память слоновья, увидят, что прежний *Amarok* возродился.

Группа разработчиков, недовольных последней реинкарнацией проекта *Amarok*, пересмотрела код его золотых дней и обновила его для лучшей работы на современных системах. Он не полностью идентичен, но близок, и если вам нравилось использовать его в те времена, вы почувствуете возврат былого. А если нет, все равно обратите на него внимание – недаром же его возвращения жаждали настолько сильно, что спустились за ним в холодную могилу контроля версий.

> «Прежний Amarok возродился для современных систем.»

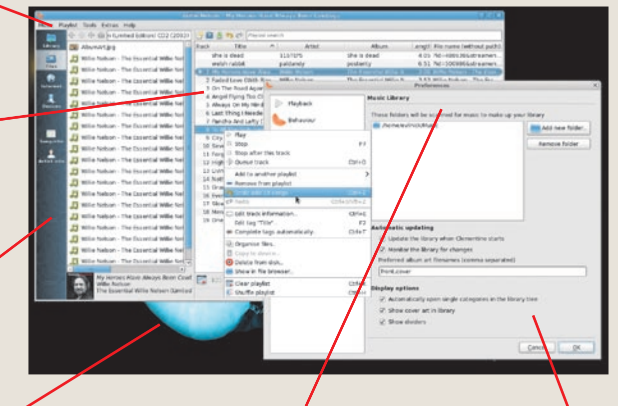


› **Искать музыку в Сети можно часами, но будьте осторожны — там встречаются ужасы.**

Хотя он очень прост в управлении, да и быстр невероятно, все же хватает штук, с которыми надо повозиться, и опций, требующих наладки. Настройки [**preferences**], к счастью, разбиты на секции для ускорения навигации; но даже так, там есть где застрять надолго. Порой кажется, что в попытках включить в него все, что можно, туда добавили довольно странные вещи. Возможность налепить сверху Гипножабу из *Футурамы* вряд ли можно считать жизненно важной функцией, но вообще-то я обнаружил, что многие треки Вилли Нельсона [Willy Nelson] здорово улучшились, когда я добавил в качестве фона ливень, и, быть может, тут есть некая сермяжная правда.

Что ни говори, *Amar*... э-ээ, *Clementine* – фантастический, быстрый музыкальный плейер, изобилующий функциями. А главное, он снабжен отличными объяснениями и поддержкой на сайте, вплоть до того, что там даже имеются преднастроенные бинарники.

Танцы под один мотив...



Меню Управление Вот увидите, пользование клавишами и всплывающими меню станет вашей второй натурой.

Панель Источников [Panel Sources] Переключение с одного источника на другой и доступ к онлайн-источникам музыки.

Контекстное меню [Context Menu] Щелкнув правой кнопкой по треку, вы получите массу опций, в том числе опцию редактирования данных в тэгах.

Обложка [Cover Art] Может быть считана с MP3, или ее можно найти. Есть также множество режимов визуализации.

Кнопки управления [Controls] Вопреки традиции, они расположены ниже главной панели.

Супер-настройка [Super Config] Вам предоставлена возможность бесстыже настраивать практически всё.

Инструмент видео

Hasciicam

Сайт http://ascii.dyne.org/

Род людской, к которому принадлежат многие наши читатели, потратил немало времени и усилий, создавая все более сложные и точные методы захвата движущихся изображений.

Тем не менее, после всех потуг нацелить на них мульти-мегапиксельные сенсоры и запечатлеть результат в широкополосном потоке данных, все равно возникает извращенное стремление превратить это все в низкокачественный анимированный текст ASCII. Видимо, потому, что это круто смотрится.

Наверное, всем знакома *AAlib*, библиотека для перевода изображений в текст, и именно она выполняет всю закулисную работу для *Hasciicam*, который просто предлагает несколько опций для контроля и управления генерированным потоком.

Здесь нет никакого GUI, *Hasciicam* работает чисто в командной строке, но не переживайте – сделать что-то полезное можно и без особых хитростей: если запустить его без всяких опций, появится новое окно с текстовым воспроизведением того, на что направлена камера.

Ну да, нужна еще и камера, причем такая, которая поддерживается в качестве стандартного устройства Video4inux, потому что там-то *Hasciicam* и будет ее искать.

Если камера у вас не одна, придется разбираться с опциями командной строки, где вы заодно обнаружите, что можно баловаться с яркостью и контрастом, настраивать нужный размер и управлять частотой кадров.

> «Сделать что-то полезное можно и без особых хитростей.»

› Что-то лучше смотрится в черно-белом отображении, а что-то — с низким разрешением.

Хотя по умолчанию эта программа извергает текст во вновь открытое окно на рабочем столе, изначальным намерением разработчиков было создание простого в использовании и экономичного по полосе способа отображать ASCII-фицированные изображения на сайтах. С этой целью файлы дополнительно сохраняются в виде огромного текста или переводятся в формат HTML, и вы можете задавать им права или даже пересылать данные напрямую через FTP на удаленный сайт.

Программа запуска

PieDock

Версия 1.2.0 **Сайт** http://www.markusfisch.de/

Нет в мире совершенства: что-то улучшить всегда можно. И это отлично – иначе мы по сию пору сидели бы в DOS. Наверное, ничто не претерпевает столько «усовершенствований», как интерфейс пользователя вашего рабочего стола Linux, и *PieDock* – еще одна попытка создать шикарный всплывающий шведский стол из приложений.

Идея его не нова, но это приложение, по крайней мере, достаточно незатейливо – оно работает и не доставляет сложностей в настройке. Запустив его, просто нажмите на определенную вами кнопку-триггер, и, где бы ни была мышь, перед вами появится колесо чудес с созвездием значков, жаждущих вашего внимания. «Пирог» анимирован, выглядит очень натуральным и ровно движется по мере того, как вы идете по кругу в поисках нужного приложения.

PieDock явно многим обязан старому модулю-доку для запуска приложений Mac OS X, но он грамотно запрограммирован и настраивается под притязания самых разборчивых пользователей.

Если вам неохота возиться с компиляцией из исходника, для быстрого знакомства с ним загляните на сайт и скачайте соответствующую блиц-версию, которая включает готовый бинарник, ресурсы и рабочую конфигурацию (хотя некоторые значки могут быть недоступны).

В файле настройки стоит разобраться – настраивается решительно все: от жизненно важных моментов (например, того, что будет являться триггером – по умолчанию это кнопка мыши при нажатой клавише «Windows») до включаемых значков, мест, где их искать, и более продвинутых опций рендеринга. Можно также настроить разные действия в зависимости от различных событий с помощью выбранного значка, так что он может стать основой для более сложной системы управления, если вы этого пожелаете.

> «В файле настройки стоит разобраться: настраивается все.»



› Клянусь головками священного сыра — это модуль запуска, и самой кольцеобразной формы!

Даже если вы вполне счастливы со своим нынешним меню и модулем запуска приложений, *PieDock* все же стоит установить – хотя бы для демонстрации неверующим.

3D-моделирование

Khayyam/floyd

Версия 04.02.2011 **Сайт** http://sourceforge.net/projects/floyd/

Движущийся палец пишет; а написав, он, возможно, загрузит 3D-модели и решит, что пара изображений – идея хорошая. За все эти годы было немало отличных приложений для 3D-рендеринга, но одним из самых революционных был *Poser*. Увы, будучи весьма популярен на платформах Mac и Windows, он никогда не переходил на Linux.

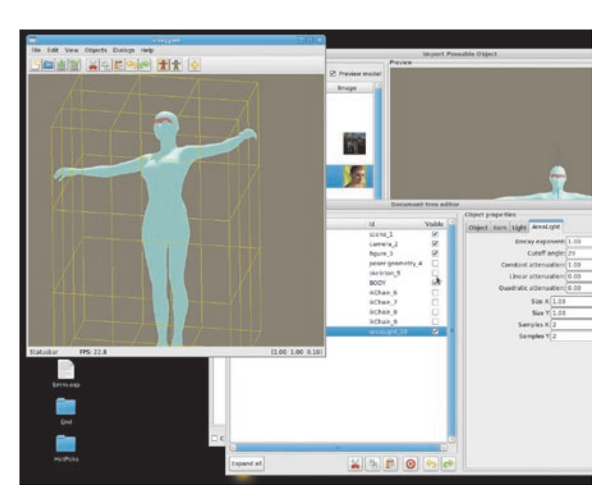
Khayyam являет собой утраченное звено – это создатель фигурок персонажей и аниматор, читающий файлы *Poser*, а также множество других объектов, позволяя вам размещать их и подвергать рендерингу.

По сути, *Khayyam* – всего лишь часть коллекции 3D-приложений, известной как *Floyd*. В него вовлечено множество общего кода и библиотек, и скомпилировать *Khayyam* отдельно нельзя без релиза *escher* (не довольно ли имен? [*Khayyam* — поэт Омар Хайям, *escher* – художник Мауриц Эшер, создававший 3D-иллюзии посредством плоских рисунков, *Floyd* – ученый-компьютерщик Роберт Флойд, – *прим. пер.*]), который подберет все необходимое.

Изрядная часть работы покамест в процессе, но даже примитивная возможность загрузить модель *Poser* и обработать ее каким-нибудь POV-Ray для некоторых будет большим скачком. Вы даже можете позиционировать модель – это если у вас есть доступные файлы *Poser*. Честно говоря, работа с моделями в *Khayyam* сейчас довольно непроста, но можно как минимум загрузить объекты, позы и материалы, добавить свет и экспортировать их для рендеринга.

Если у вас нет *Poser*, не переживайте. Компания под названием DAZ прославилась созданием моделей – альтернатив Poser. И теперь они доступны, многие – бесплатно, на сайте Daz (www.daz3d.com). Чтобы загрузить их, придется помучиться с *Wine*, но сам *Poser* не потребуется: воссоздадутся соответствующие директории *Poser*, пригодные для чтения в *Khayyam*.

> «Создатель персонажей и аниматор, читающий файлы Poser.»



› Тонкого контроля и простого интерфейса *Poser* здесь нет, но *Khayyam* умеет загружать, обрабатывать и выводить модели.

Редактирование видео

LiVES

Версия 1.4.2 **Сайт** http://lives.sourceforge.net/

Найти реально работающее приложение для редактирования видео в Linux раньше было непростой задачей. Ныне ситуация несколько улучшилась благодаря приложениям вроде *Kdenlive* и *OpenShot*, недавних героев HotPicks. И еще есть LiVES – это статья особая.

LiVES пригоден как традиционный оффлайн-редактор. В этом режиме достаточно функций, включая мультитрековое редактирование, аудио- и видеоэффекты, переходы и весь обычный антураж. А выделяется он тем, что умеет работать еще и с прямой трансляцией видео.

Это не только облегчает задачу монтажа в студии – это также значит, что программа способна работать «видеожокеем» в режиме live, микшируя предварительно записанный видеоматериал с «живым», причем из многочисленных источников.

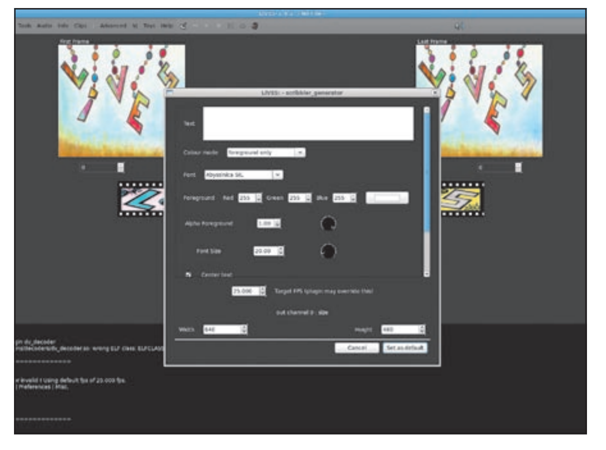
Несмотря на изобилие функций, в работе с базовыми *LiVES* не сложнее своих коллег, и в нем широкий выбор эффектов, от рядовых до довольно эклектичных. Если вы умеете делать видео в Linux, вы, вероятно, освоитесь и в *LiVES*.

Интерфейс пользователя, возможно, несколько непривычный, и нельзя отрицать, что при желании зайти чуть дальше склеивания пары клипов придется потрудиться, чтобы разобраться со всеми опциями – даже простое чтение всех пунктов меню займет несколько часов.

Наградой вашему упорству будет мощная система практически на все руки. Если какой-то функции не окажется – ну, напишите модуль расширения сами...

Единственный недостаток *LiVES* – пока не добитые капризы при компиляции. В случае, если у вас она почему-либо не сработает, обратитесь к live-дистрибутивам типа Dynebolic (http://dynebolic.org/) или AvLinux (http://www.bandshed.net/AVLinux.html).

> «Не окажется функции, напишите модуль расширения сами.»



› *LiVES* умеет работать с прямой трансляцией сразу из нескольких источников.

Инструмент безопасности

Nixory

Версия 1.0.3 **Сайт** http://sourceforge.net/projects/nixory/

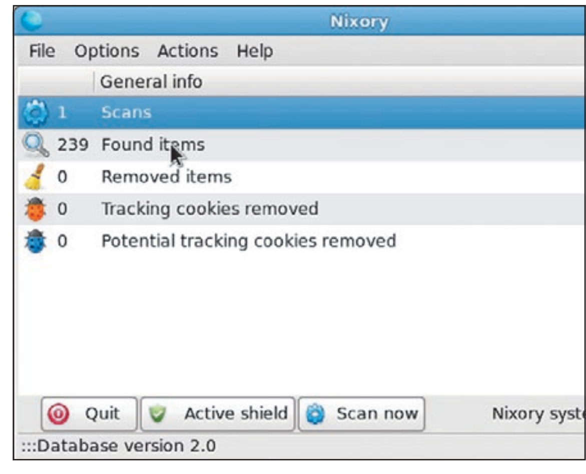
Убеждение, что в Великобритании всех подряд по 300 раз на дню снимают камерами видеонаблюдения, как оказалось, сильное преувеличение. Забавно, что все камеры, кажется, направлены как раз в обратную сторону, когда кто-то, например, пытается угнать мой мотоцикл. В любом случае, не так уж за нами и следят, как мы воображали. Я неправ?

Куда большую угрозу вашей свободе создает вытягивание из вашего браузера личных данных при каждом заходе в Сеть. В отличие от видеонаблюдения, нет никаких законов или ведомств, контролирующих этот процесс.

Cookies с виду безобидны, и нередко так и есть, но они бывают и агентами Вельзевула: именно эти невинные крошки помогают злобным админам отслеживать, куда вы заходите, что читаете, с кем общаетесь... и натравливать на вас «целевую рекламу». Поэтому-то, когда команда LXF заходит на один и тот же сайт, Грэму рекламируют шлепанцы, мне – шампунь, Эффи получает чимичанги, а Майк видит... молчу: всему есть предел.

Nixory – не просто модуль расширения для браузера, спасающий вас от вредоносной глупости cookies; это cookie-монстр совершенно иного толка. Как отдельная программа, он нацелен на разные браузеры – например, *Chrome* и *Firefox*; хотя нетрудно расширить его и до других.

Nixory написан на Python и имеет простой GUI. Конечно, дай вам волю, вы вызовете его пару раз и забудете; вот почему *Nixory* предусматривает также режим демона «Активный щит» – постоянный сервис, проверяющий и удаляющий дурные куки даже тогда, когда браузер на ходу.



› ***Nixory* охотится на плохие cookie в фоновом режиме, без вашего участия.**

> «Cookies с виду безобидны, но бывают агентами Вельзевула.»

Только что вышло обновление кода *Nixory* – на него стоит взглянуть; но оно опирается на новый движок базы данных и требует свежего исходника *SQLite* – могут быть сложности с его компиляцией.

Менеджер изображений

Shotwell 0.9

Версия 0.9 **Сайт** http://yorba.org/shotwell/

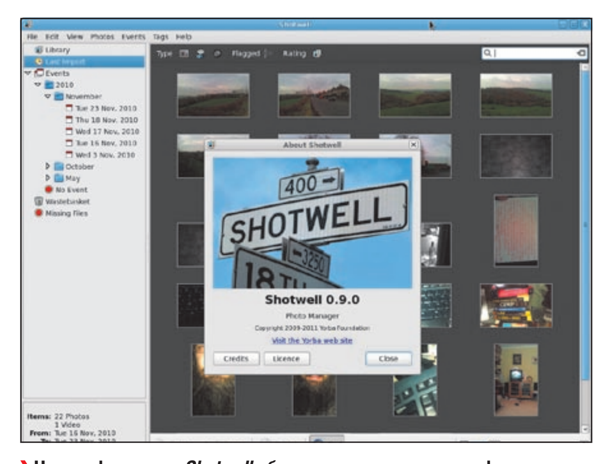
Shotwell – отличный выстрел [по-англ., – *прим. пер.*], э-ээ, прямо в десятку – ведь его выбрали на замену более мощным, но, возможно, тяжеловатым приложениям для управления фотографиями в таких популярных дистрибутивах, как Fedora и Ubuntu. Идея была в том, что большинство просто хотят запихнуть свои фотографии очаровательных котят и детишек с кастрюлями на головах (или наоборот) себе в компьютер, а оттуда перегнать их на сияющий простор социальных сетей.

Shotwell – очень простой в использовании инструмент, в первую очередь благодаря тому, что и задачи его просты. Меньше опций – меньше кнопок, где можно запутаться, а значит, меньше ошибок.

Однако у разработчиков в их большом общем рукаве завалялось несколько неожиданных трюков, поскольку в этой свежайшей версии появились – надо же! – новые функции.

Новое в релизе 0.9 – поддержка изображений TIFF, улучшение работы с изображениями формата RAW и возможность печатать несколько фотографий. Забавно, что последней функции пришлось ждать выхода 0.9, чтобы туда пробраться, но, конечно, люди уже не столь часто печатают фотографии, как прежде, и вполне понятно, что программа учитывала нужды потребителей.

Функции поиска значительно улучшены, что очень важно. Как известно всем владельцам цифровых камер, емкие карты памяти и возможность фотографировать, сколько хочешь, означают, что обычно на каждое мелкое событие набирается до пары десятков изображений, и возможность быстро выбрать достойные (вы же назначаете им рейтинг, да?) или отыскать по текстовым полям очень приветствуется.



› **Новые функции в *Shotwell* облегчают поиск хороших фото среди вороха отснятой вами дряни.**

> «Меньше опций – меньше кнопок, где можно запутаться.»

Поклонники *Shotwell* не должны отчаиваться – все эти функции не слишком затруднят работу пользователя, так что *Shotwell* останется избранным менеджером изображений на большинстве настольных ПК. Компиляция из исходника вполне обычная, но затрагивает столько новых библиотек, так что, возможно, проще будет подождать его появления в репозитории вашего дистрибутива.

HotGames Развлекательные приложения

Стратегия в реальном времени

Gigalomania

Версия 0.16 **Сайт** http://homepage.ntlworld.com/mark.harman/comp_gigalomania.html

Игра *Mega Lo Mania* была изначально запущена на **Amiga** в 1991, и создали ее настоящие безумцы из Sensible Software. Как и все игры того времени, она не отличалась особой оригинальностью – в конце концов, это всего лишь еще одна игра в Бога – но что-то было захватывающее в смешении разных уровней, моделях игры в реальном времени и, конечно, в божественной власти. Sensible, наверное, никогда не создавала скучных игр, так что попытка воссоздать их гений является довольно смелой.

Gigalomania, однако, пытается стать лучше аж на порядок. Здесь практически те же самые элементы игры. Вам нужно отыскивать технологии разных эпох и экипировать своих соратников орудиями, позволяющими им побеждать других, не столь богоподобных существ, конкурирующими за ту же карту мира.

Придраться тут есть к чему, и игра, вероятно, все же не дотягивает до планки оригинала, но проделанная работа впечатляет. Хотя разработка находится в активной стадии, играть вполне можно, и для упражнения ваших мозгов уже имеется 28 уровней. Основы игры схватываются буквально за пару минут, а на сайте проекта есть полезное руководство на случай, если вы где-то застрянете.

Правда, установка игры несколько запутанная. Она разработана для Windows или Linux благодаря магии SDL, однако бинарников для Linux нет – вам придется компилировать из исходника. К сожалению, игра использует стороннюю библиотеку, хоть и свободную от лицензионного маразма, но все же недоступную для распространения через **LXF**, и вам придется загрузить ее самостоятельно. Это библиотека *Fmod* (http://www.fmod.org/) – установить ее очень легко, коль скоро вы ее добудете.

> «Основы игры схватываются за пару минут.»



› Скоро все эти пиксели будут мои! А все эти подпроцессы станут поклоняться мне, мне одному!

Удачи вам в освоении островов, и пусть вам сопутствуют ваши боги.

Еще одна стратегия в реальном времени.

Zod Engine

Сайт http://sourceforge.net/projects/zod/files

Итак, старая добрая классика дарит нам в редакции **LXF** еще один месяц удовольствий. Древняя игра с лаконичным именем *Z* была крупным успехом Bitmap Brothers в конце 90-х и у многих изменила воззрение на работу стратегии в реальном времени. Тогда она казалась мозголомным отрывом от тех игр типа RTS, в которые мы играли; ну, а сегодня, конечно, выглядит чуть старомодной.

Проект *Zod Engine* работает уже несколько лет, и его цель – не только достоверно воспроизвести *Z*, но и заново открыть его нашему времени. А это требует значительных усилий. Для особо ностальгирующих имеются уровни самой *Z*, которые можно проходить по порядку, но без видеоэпизодов оригинала.

Игра допускает до восьми игроков, включая управляемых компьютером персонажей AI. Нужно изрыгать из вашей базы боевые отряды и слать их в координированные атаки на врага. Игроки AI не слишком расторопны – лучше поищите партнеров онлайн. *Zod* построена по схеме клиент/сервер, так что будет работать и через LAN, и через Интернет. Игра потребляет трафик весьма скромно и вас не разорит.

Авторы также включили программу запуска с GUI – она поможет вам освоиться и сохранить ваши любимые настройки для быстрого восстановления игры. Не обошлось без огрехов (и без библиотек *wx*, так что понадобится установить и их тоже), но зато не нужно запоминать полдюжины опций командной строки.



› Ага, это 90-е? Тогда займемся-ка мы поджигательством...

> «Нужно изрыгать боевые отряды и слать их в атаки.»

Инструмент отмены удаления

giis for ext4

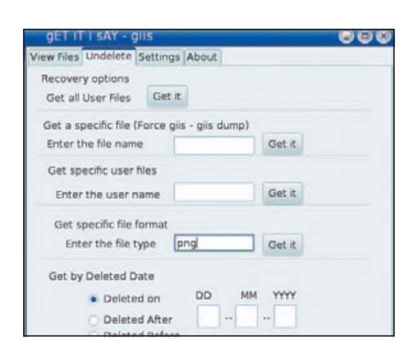
Версия 0.6 **Сайт** http://www.giis.co.in/

Giis – странноватый акроним для довольно полезного инструмента файловой системы. *Get It I Say* [Получи, сказано тебе, – *прим. пер.*] задуман для восстановления файлов на разделах **Linux. Он не делает ничего особо крутого**, но умеет восстанавливать файлы, когда капитулируют другие способы, если только у вас нет шестнадцатеричного редактора, точного знания узлов inode файловой системы и неиссякаемого терпения.

Внесем ясность: чудес эта программа не творит. Она не воскресит те милые сердцу фото, что вы удалили пять лет назад, и не спасет ваши файлы с горящего жесткого диска. Зато она отменит операцию удаления – если вы грохнули файлы, удалению не подлежащие (**rm -rf *** не в той директории? Не краснейте: с кем не бывает...), их можно будет спасти, если только поверх них уже не записались другие.

Итак, *giis* не очень-то предотвращает потерю данных, но это отличный инструмент на случай, если вы по неосторожности удалили нечто важное. Программа работает по принципу мгновенных снимков структуры диске через определенные промежутки времени в фоновом процессе.

Хотя изначально *giis* был инструментом командной строки, у него имеется GUI, но, честно говоря, проще будет использовать оболочку для восстановления файлов. Есть несколько разных версий *giis*. Самая последняя поддерживает файловую систему ext4, но есть версии и для файловой системы ext3.



› Этот инструментик не творит чудес, но спасет вас, если вы вдруг обдернетесь.

Виртуализация

Virt Manager

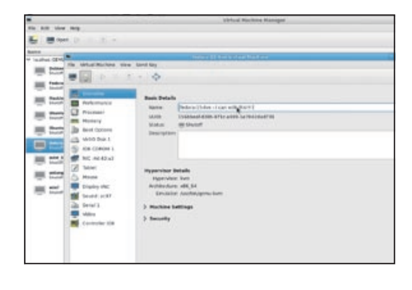
Версия 0.8.7 **Сайт** http://virt-manager.org/

Когда-то виртуальные машины казались плодом больного воображения и результатом злоупотребления энергетическими напитками, но это было давно. Фактически, впервые идея виртуализации была реализована на практике в 60-е, на IBM, сначала как возможность для пользователей делить время CPU, а потом – как способ разработки мейнфреймов: было проще и дешевле тестировать виртуальное оборудование, чем создать всю машину, а потом убедиться, что она не работает.

На настольных системах в виртуализации раньше доминировала *VMware*, но постепенно проекты с открытым кодом отняли у нее славу. *Virtualbox* – сильный конкурент, но его будущее под вопросом, поскольку он угодил под «заботливое» крыло Oracle.

К счастью, есть свободные альтернативы. *Virt Manager*, финансово поддерживаемый Red Hat – очень удобный способ управлять виртуальными устройствами *KVM/Qemu* и *Xen* с вашего настольного ПК. Он работает настолько хорошо, что фактически, без его помощи компиляция этой части журнала заняла бы, вероятно, месяца два, а так на нее ушло, кхе, полтора.

Последняя версия расширила функциональность SPICE, системы для имитации работы ВМ в сети. Это интересно скорее разработчикам систем и администраторам, чем обычным пользователям настольного Linux, но все же с *Virt Manager* стоит поэкспериментировать. LXF



› Ряд исправлений в интерфейсе позволяет внести изменения, не переустанавливая всю машину целиком.

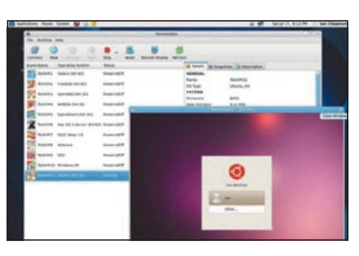
Также вышли

Новые и обновленные программы, тоже достойные внимания...

WEBGUI 7.10.13
Автопостроение сайтов с CMS для тех, кому лень программировать.
http://www.webgui.org/

GTK 3.08
Библиотеки, работающие на половине Linux-машин, получили долгожданные обновления.
http://www.gtk.org/

RemoteBox 0.8
Рулите по сети виртуальными машинами *Virtualbox* через этот удобный GUI.
http://remotebox.knobgoblin.org.uk/



› Теперь можно не только совать нос в чужой удаленный рабочий стол, но и влезть в чужую ВМ.

FreeSeer 2.0
Видеосъемка, которая стала простой. Или, на худой конец, проще.
https://github.com/fosslc/freeseer/wiki/

TEA 29.0.0
Постоянный посетитель наших страниц, этот быстрый текстовый редактор в данном основном релизе улучшил графику.
http://tea-editor.sourceforge.net/

gcc 4.6.0
Та самая первородная протоплазма – коллекция компилятора Gnu, оптимизированная и с рядом важных изменений.
http://gcc.gnu.org/

Conquests 0.11
Многообещающий клон *Civilization*. Пошаговая стратегия, возможно, вернется...
http://homepage.ntlworld.com/mark.harman/conquests.html



› В своих элегиях я помяну все преступления Гелиогабала.

На диске

Дистрибутивы, приложения, руководства, подкасты и более того...



Прочтите всю информацию, необходимую для использования DVD! Если вы новичок в Linux, откройте файл index.html на диске и перейдите в раздел Справка, где вы найдете руководства по вопросам, перечисленным справа:

Майк Сондерс, редактор диска
mike.saunders@futurenet.com

- Что такое Linux?
- Что такое дистрибутив?
- Загрузка ПК с DVD
- Разбиение жесткого диска на разделы
- Навигация по файловой системе
- Использование командной строки
- Установка программ

Дистрибутивы Linux

Gnome 3, Arch, Jolicloud

Революция, а не эволюция. Вот такой менталитет у Gnome 3, и нигде это не заметно столь явно, как в дизайне рабочего стола. Новая оболочка Gnome 3 подорвала все основы дизайна GUI, выкинув старые панели и меню приложений и заменив их броским интерфейсом с упором на задачи, где все крутится вокруг кнопки Activities. Выбор крайне спорный – подробнее об этом читайте в статье «Gnome 3.0 с нами!».

Но вам не только покажут симпатичные картинки: с Gnome 3 можно познакомиться сию же секунду в режиме live с **LXFDVD**. Запустите компьютер с диском в приводе, и оболочка Gnome 3 запустится из загрузочного меню. (Учтите, что она из альфа-версии Fedora 15 и требует аппаратной поддержки 3D – иначе вы получите старый рабочий стол Gnome. Если возникнут проблемы, форум www.fedoraforum.org должен помочь.)

Далее идет Arch Linux – дистрибутив, малоизвестный вне круга компьютерных

> «Замените старые панели и меню броским интерфейсом.»

гениев. Arch стремится к легковесности и простоте, концентрируясь на «элегантности, правильности кода и минимализме». Он, безусловно, не для новичков, и вам не обойтись без солидных познаний в Linux, чтобы с ним справиться, зато предлагает изрядную долю гибкости, и вы склепаете именно тот Linux, который вам нужен. Вы найдете ISO-образ самого свежего основного релиза в разделе **Distros LXFDVD**, который идет рука об руку со статьей об **Arch Linux в разделе «Учебники»**. Образ можно записать на CD-R и загрузиться с него, или поиграть с ним в эмуляторе ПК – например, *VirtualBox*.

А последний из наших дистрибутивов – Jolicloud, ОС на базе Ubuntu; ее специальность – облачные вычисления [cloud computing]. Раз уж мы проводим все больше времени в приложениях, предлагаемых браузерами, возникают и дистрибутивы, содействующие этому виду работы на компьютере. Узнайте об этом больше из нашего обзора – на диске имеется ISO, который можно перенести на USB-брелок, следуя инструкциям в **INSTALL.TXT**.



Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.



› Gnome 3 с его новой оболочкой резко отходит от предыдущих релизов.

Другое ПО

Firefox, wxHexEditor...

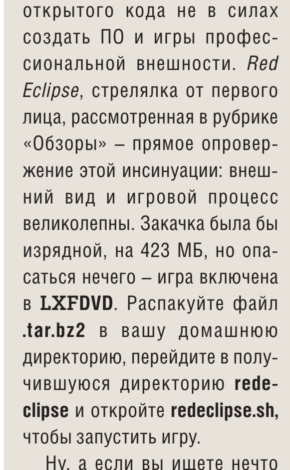
Команда *Firefox* объявила об уплотнении графика выхода релизов – ко времени выхода журнала, чего доброго, этак прибудет уже *Firefox 18*; но вряд ли там будет особо много новых функций. Зато *Firefox 4* – серьезное обновление второго в мире по популярности браузера; читайте об этом в нашем обзоре. А вот как его установить. В разделе Internet/Firefox на DVD скопируйте **firefox-4.0.tar.bz2** в домашнюю директорию. Затем распакуйте его через свой файловый менеджер и запустите **firefox** в полученной директории – или, альтернативно, проделайте все это в командной строке:

```
tar xfvj firefox-4.0.tar.bz2
firefox/firefox
```

Затем можно создать значок запуска для бинарника **firefox** внутри директории **firefox** для выбранного вами менеджера окон или рабочего стола. Вы найдете настройки и кэш *Firefox* в **.mozilla/firefox/** в вашей домашней директории.

Для разработчиков и низкоуровневых ремонтников мы припасли отличный шестнадцатеричный редактор – а именно, *wxHexEditor*. Конечно, такие редакторы в мире открытого ПО идут по дюжине на грош, но туз в рукаве *wxHedEditor* – его способность работать с большими файлами. Если вам вздумается подвергнуть шестнадцатеричному редактированию свой жесткий диск через его узел устройства в **/dev**, *wxHedEditor* сделает это, не дрогнув. Как видно из названия, он применяет в своем интерфейсе *wxWidgets*, так что нужно будет установить эту библиотеку.

И напоследок: при всей нашей любви к Linux, иногда приходится запускать и сугубо Windows-программы. Долгие годы нас тут выручала *Wine*, но обеспечивать совместимость приходится как-никак с весьма путаной ОС, и иногда один только запуск **wine некаяпрограмма.exe** не срабатывает. На наше счастье, в подобных случаях всегда готов помочь *Winetricks*, до отказа упакованный настройками для игр и приложений. Он умеет скачивать программы, устанавливать недостающие DLL, применять разнообразные обходные приемы и проделывать всякие другие фокусы, чтобы обеспечить максимальную совместимость.



› Не идите на поводу у сложности *Wine* — *Winetricks* устанавливает программы Windows легко.

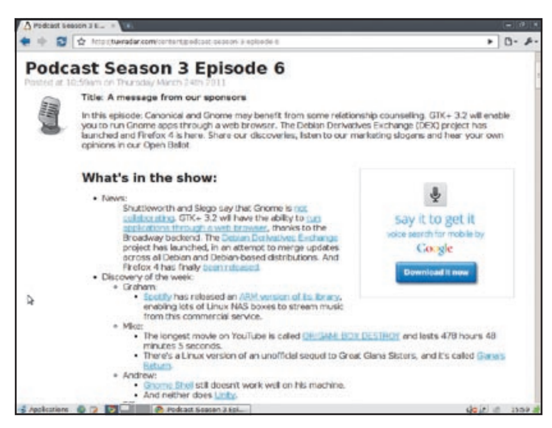
Дополнительные материалы на диске

Код, подкасты и документация

Наше новое приложение по программированию просто лопается от смачных фрагментов кода, достойных пробы, но, к счастью, минули времена журналов для 8-битных компьютеров, когда приходилось набивать строчку за строчкой в BASIC, и весь код можно взять в разделе Magazine на диске.

А если кто не слышал, команда Linux Format каждые две недели записывает подкаст для нашего братского сайта www.tuxradar.com. Мы собираемся у микрофона, беседуем о самых насущных проблемах в мире открытого кода, а иногда отвлекаемся на страсть Эффи покушать чимичангу. Все это очень здорово, и вы можете прослушать самые последние серии подкастов в разделе Magazine/Podcast на диске.

И, наконец, для тех, кто только-только вступает в светлый мир Linux, у нас есть подборка мини-учебников, дружелюбных к начинающим. Откройте **index.html** на диске, выберите пункт меню «Новичок в Linux» вверху слева, и перед вами предстанут руководства по терминологии Linux, загрузке дисков, установке программ и т. п. Там также есть указатели на полезные онлайн-ресурсы, которые всегда вас спасут в случае проблем.



› Подкаст мы делаем раз в две недели — спешите в раздел Magazine/Podcast!

И это еще не все!

Четыре классных игры

Не верьте, что сообщества открытого кода не в силах создать ПО и игры профессиональной внешности. *Red Eclipse*, стрелялка от первого лица, рассмотренная в рубрике «Обзоры» – прямое опровержение этой инсинуации: внешний вид и игровой процесс великолепны. Закачка была бы изрядной, на 423 МБ, но опасаться нечего – игра включена в **LXFDVD**. Распакуйте файл **.tar.bz2** в вашу домашнюю директорию, перейдите в получившуюся директорию **redeclipse** и откройте **redeclipse.sh**, чтобы запустить игру.

Ну, а если вы ищете нечто менее стремительное, тогда попробуйте *Hedgewars*, *Worms*-подобную игру. Мы рассматривали *Hedgewars* не раз: она часто получает прекрасные обновления. Последний на данный момент релиз 0.9.15 вводит новое оружие и возможность создавать свои карты.

Затем идет *Berusky*, логическая игра на основе головоломки классического жанра *Sokoban*. Прохождение всех ее 120 уровней, очень тщательно продуманных, займет вас надолго. И наконец, у нас есть обновление для *Legend of Edgar*, занимательной двумерной платформенной игры, к которой в этом релизе 0.8 добавился целый раздел.



› *Hedgewars* — как *Worms*, только действуют здесь ежи. Мило!

На диске

Дайте Linux полную свободу — и вас похвалит РМС!

Дистрибутив Linux

Trisquel GNU/Linux 4.5

Вот вам новость, которая, возможно, вас шокирует: вы, вероятно, используете не вполне свободное ПО, по крайней мере, в понимании «свободы как слова». Удобства ради, многие настольные дистрибутивы Linux включают фрагменты проприетарного кода – например, прошивки драйверов; в итоге они свободны не целиком, а лишь в основном.

Если свойство свободы Linux вас не сильно волнует, то проблемы нет; но если вы пришли в Linux в поисках свободы от проприетарных оков и контроля корпораций, то это вопрос важный.

Но решение есть. Trisquel GNU/Linux – дистрибутив Linux, в котором свободен каждый байт: проприетарного ПО здесь ни на йоту.

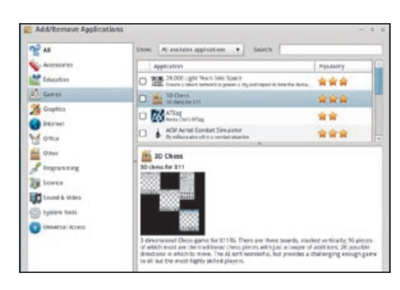
Он одобрен Фондом свободного ПО [Free Software Foundation] и предоставляет самый чистый способ работы в Linux, позволяющий вам не марать рук закрытым кодом. Дополнительный бонус – он основан на жутко популярном Ubuntu 10.10, и ему доступны все обширные репозитории программ (свободного ПО) в **Ubuntu** и Debian, и большинство документации в сети тоже к нему применимо.

Trisquel GNU/Linux 4.5 загружается в режиме live и устанавливается прямо с DVD этого месяца: просто запустите свой компьютер с установленным в приводе DVD и выберите Trisquel в меню. (Если ваша машина загружается с жесткого диска, вам, возможно, придется изменить порядок загрузки в BIOS – более подробную информацию см. в документации вашего компьютера.) Рекомендуемые минимальные системные требования для гладкой работы: 1-ГГц x86 CPU (можно 32-битный или 64-битный), 512 МБ ОЗУ и 10 ГБ места на жестком диске.

В режиме live, непосредственно с DVD, Trisquel работает намного медленнее, чем после установки на жесткий диск, так что пусть его скорость не вводит вас в заблуждение. Режим live – просто хороший способ увидеть, поддерживает ли он ваше оборудование, и познакомиться с наличными программами (щелкните по голубому значку в нижней левой части экрана для навигации по установленным программам). Trisquel включает все, чего можно ожидать от современной настольной ОС – интернет-приложения, офисный пакет, программы работы с графикой и многое другое.

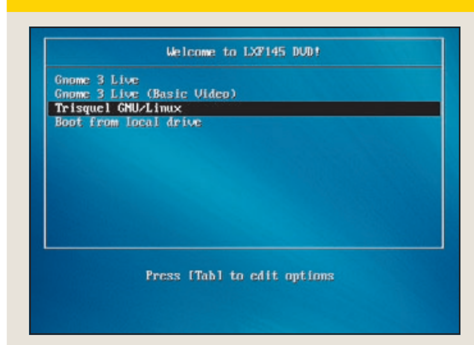
Примечание по разбивке диска на разделы для новичков в Linux: как и любой операционной системе, Linux нужно место на жестком диске. Он может сосуществовать с уже имеющимися на нем ОС – например, Windows; и после установки у вас появится загрузочное меню, позволяющее выбрать ОС при загрузке вашего ПК. Программа установки сожмет порцию диска Windows; это процедура безопасная, но все же рекомендуем сначала сделать резервные копии важных данных – мало ли что!

› Через голубой значок внизу справа на экране входа можно активировать инструменты доступа для пользователей-инвалидов.



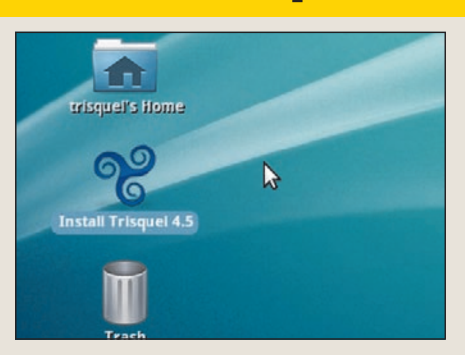
› Щелчок по значку Trisquel > Add/Remove Applications дает тысячи дополнительных приложений из Интернет.

Шаг за шагом: Устанавливаем Trisquel GNU/Linux



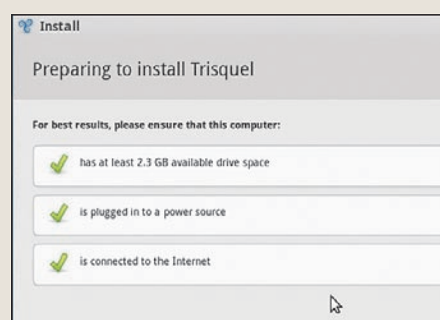
1 Загрузка

Загрузите свой ПК с **LXFDVD**, в меню загрузки выберите пункт Trisquel GNU/Linux клавишами курсора и нажмите Enter.



2 Рабочий стол

Перед вами появится рабочий стол – приступайте к знакомству. Для запуска процесса установки дважды щелкните по значку Install на рабочем столе.



3 Программа установки

Загрузится программа установки и проверит, хватает ли у вас свободного места на жестком диске, чтобы разместить операционную систему.

Как бы мне?..

Если вы только начинаете работать в Linux, то после установки Trisquel вы можете испытывать неуверенность насчет выполнения некоторых действий. Вот список наиболее общих задач и способов их решения...

» Работа в сети Щелкните по значку с голубым глобусом на нижней панели, чтобы запустить обновленную версию *Firefox*, второго в мире по популярности браузера с сотнями расширений.

» Работа с файлами Щелкните по значку с голубой папкой на нижней панели, чтобы сохранять и получать доступ к своим личным документам.

» Воспроизведение музыки и видео Щелкните по значку Trisquel внизу слева и перейдите в подменю Sound & Video, чтобы попробовать музыкальный плейер *Exaile* и Movie Player для фильмов.

» Редактирование изображений В меню Trisquel перейдите в Graphics и попробуйте *GIMP*, всестороннюю программу редактирования изображений; у нее много общего с *Adobe Photoshop*.

» Чат онлайн В меню Internet вы найдете *Pidgin*, блестящую программу для быстрого обмена сообщениями, которая поддерживает все популярные протоколы – MSN, Jabber и Google Chat.

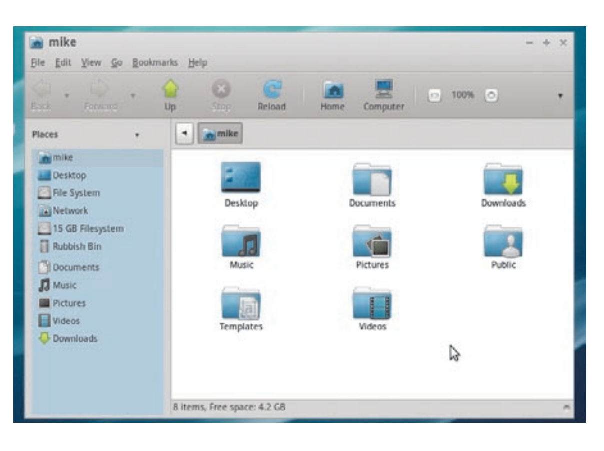
» Настройка системы Перейдите в меню System, а затем – в подменю Preferences и Administration; там множество инструментов для работы с вашей системой.

» Поиск новых программ Простая утилита – **Add/Remove Applications**; а если вам нужен более продвинутый инструмент, попробуйте System > Administration > Synaptic.

» Выход и выключение Щелкните по логотипу Trisquel и затем – Shut Down, чтобы выйти из системы или выключить компьютер.

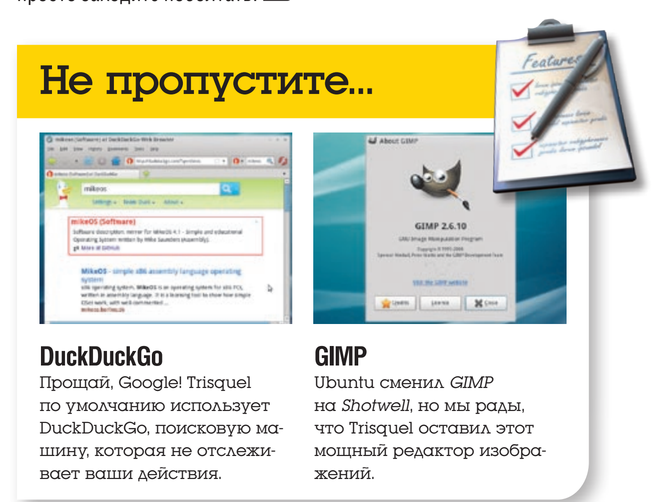
Сайт Trisquel – http://trisquel.info; там есть ссылки на документацию (http://trisquel.info/en/wiki) и форумы (http://trisquel.info/en/forum/users). Эти ресурсы весьма полезны, но стоит принять во внимание и происхождение дистрибутива от Ubuntu и Debian: сайты этих дистрибутивов помогут получить подсказку и пообщаться с другими пользователями. По правилам сетевого этикета, следует ясно и точно сформулировать свой вопрос, предоставив информацию по сообщениям об ошибках, настройке оборудования или иную соответствующую информацию.

А наши форумы сидят на сайте www.linuxformat.com/forums/. Мы – очень дружелюбная компания, и дистрибутивы используем самые разные; регистрация у нас свободная, и мы не рассылаем спам и тому подобное. Пусть у вас и нет вопросов по Trisquel – просто заходите поболтать! LXF



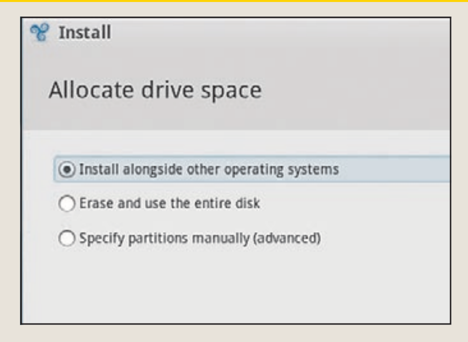
› Trisquel – один из самых миловидных дистрибутивов среди нам знакомых, с мягкими оттенками и приятными виджетами.

Не пропустите...

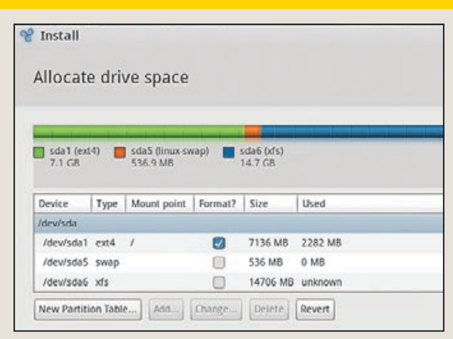


DuckDuckGo
Прощай, Google! Trisquel по умолчанию использует DuckDuckGo, поисковую машину, которая не отслеживает ваши действия.

GIMP
Ubuntu сменил *GIMP* на *Shotwell*, но мы рады, что Trisquel оставил этот мощный редактор изображений.



4 Разбиение на разделы
Теперь можно изменить размер имеющегося раздела Windows (если он есть), занять весь жесткий диск полностью или поделить его на разделы вручную (это для экспертов).



5 Ручное разбиение
Если вы делите диск вручную, создайте раздел root (/) в формате **ext4** с минимальным размером 5 ГБ, и раздел подкачки [swap] размером не менее 512 МБ и не более 2 ГБ.



6 Подтверждение
Когда файлы скопируются, вам зададут другие вопросы по настройке. При создании учетной записи пользователя помните о регистре! И перезагружайтесь уже в Trisquel.

Информация о диске

Внимательно прочтите это перед тем, как использовать LXFDVD!

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

» **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
» **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
» **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
» **имя_программы-1.0.1.tgz** – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;
» **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
» **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
» **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
» **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
» **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Июнь 2011 LXFDVD 145

LINUX FORMAT

ubuntu 11.04

Весенний релиз 2011 года

Сборка для i386

» Ядро 2.6.38 » Графическая оболочка Unity » Nautilus 2.32.2 » LibreOffice 3.3.2

А ТАКЖЕ: Firefox 4.0 и самые свежие версии остальных пакетов

Июнь 2011 LXFDVD 145

LINUX FORMAT

archlinux 2010.05

Дистрибутив, в котором все можно настроить самостоятельно

Плюс:
» **Gnome 3** Live CD на базе Fedora 15 alpha
» **Joilcloud 1.2** Новая версия дистрибутива для работы с «облаками»
» **Trisquel 4.5** Дистрибутив, «свободный как слово»

А ТАКЖЕ: Firefox 4.0, Mono, gCAD3D, проигрыватели музыки, игры и многое другое...

DVD-приложение к журналу Linux Format №6 (145). Июнь 2011

» Содержание

LINUX FORMAT

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИИ? Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disks@linuxformat.ru

Тираж изготовлен ООО «Уральский электронный завод». 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАФ 77-15.



Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.



Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Проект «Linux – школе»

Переход учебных заведений на свободное ПО требует не только соответствующих программных средств, но и подготовки преподавателей.



Наш эксперт

Александр Казанцев
К. т. н., доцент кафедры информатики Глазовского государственного пединститута, руководитель проекта EduMandriva, автор локализаций (и разработок) для Mandriva, *Klavaro*, *Gambas2* и *LXDE*.

Чтобы обеспечить грамотную реализацию подготовки, издательство БИНОМ совместно с «ГНУ/Линуксцентром» (ЗАО «Мезон.РУ») создали совместный проект, в котором региональные структуры повышения квалификации работников образования могут подключиться к сотрудничеству на долгосрочной основе в рамках включения в сферу образования открытого программного обеспечения Линукс. Сотрудничество предусматривает проведение целевого обучения учителей использованию СПО на основе сформированной в регионе инфраструктуры работы со школами и предложенных «Линуксцентром» и издательством БИНОМ учебных материалов. Также в этом проекте непосредственно участвуют наш журнал и образовательный проект EduMandriva.

Что же предлагает данный проект? Согласно сайту http://metodist.lbz.ru/nio/linux.php, «В связи с внедрением в школы свободного программного обеспечения (СПО), компания «ГНУ/Линуксцентр» (www.linuxcenter.ru) предоставляет на безвозмездной основе всем образовательным учреждениям (школам, УНПО, учреждениям начального и среднего профессионального образования, ИПК работников образования) комплект СПО на правах лицензии GPL. Установленные лицензией GNU GPL правила распоряжения правами для любых версий программных продуктов комплекта СПО отвечают отечественным требованиям в отношении как физических лиц (пользователей), так и юридических лиц (организаций)».

> «Проект нацелен на создание сети региональных центров.»



Текущий комплект СПО.



В настоящий момент состав комплекта СПО «Linux – школе» следующий:

- 1 CD + 1 DVD дистрибутива EduMandriva
- CD диск свободного ПО для Windows – FreeWin
- LinuxCenter DVD с методическими материалами
- Архив журнала Linux Format

Комплект СПО «Linux – школе» от компании «ГНУ/Линуксцентр» и лицензия GPL к нему предназначены для неограниченного копирования и распространения в образовательных учреждениях, а также для копирования и личного использования педагогами и учащимися, а техническая поддержка, сетевое консультирование пользователей, документальное сопровождение и своевременное обновление версий СПО осуществляется в соответствии с лицензионным договором, приобретаемом в компании «Линуксцентр».

Также проект нацелен на создание сети региональных консультационных центров Linux, так как в рамках сотрудничества издательства «БИНОМ. Лаборатория знаний» с компанией «Линуксцентр» предусмотрено обучение тьюторов ИПК на безвозмездной основе в дистанционном режиме, формирование регионального консультационного центра «Linux – школе» и передача ПО Linux для свободного распространения в школы территории.

С 2010 году компания «ГНУ/Линуксцентр» выдает сертификат «Региональный консультационный «Линукс-центр» совместно с Пакетом СПО «Linux – школе» тем учреждениям дополнительного профессионального образования работников образования, которые завершили обучение тьюторов в компании «ГНУ/Линуксцентр». В настоящий момент общее число обученных тьюторов превышает 70 человек, а география регионов простирается от Архангельска до Астрахани и от Ленинградской области до Хабаровского края, и их число постоянно растет.

Также в рамках проекта подготовлен и размещен цикл видеолекций (http://metodist.lbz.ru/content/video/kazantcev.php), по-



› **Сертификат преподавателя.**



› **Сертификат консультационного центра.**

священный операционной системе Linux. Приводим затронутые в курсе темы:

ТЕМА 1: Операционная система Linux. Проект «Linux — школе»:

» Операционная система Linux.
» Проблемы и решения при установке Linux.
» Linux и существующие образовательные решения (ЦОР, ЭОР, УМК).
» О проекте «Linux – школе» и курсах Linux.

В данной лекции рассматриваются вопросы использования операционной системы Linux: рекомендуемые дистрибутивы, вопросы технического и административного характера, с которыми можно столкнуться при установке СПО в учебном заведении. Обсуждается проект «Linux – школе», образовательная инициатива компании «ГНУ/Линуксцентр» и проект Edumandriva. Будут рассмотрены вопросы построения своих образовательных курсов и участия в программе «Региональные консультационные центры».

Методические материалы к лекции были размещены в нашем журнале ранее:

» Школа **LXF**: Школьные дистрибутивы (лето 2010) www.linuxformat.ru/download/LXF134.school.pdf
» Школа **LXF**: Укрощение железа www.linuxformat.ru/download/LXF141.school.pdf

ТЕМА 2: Использование СПО в школьном курсе «Информатика и ИКТ»:

» Применение свободных программных средств при изучении информатики и ИКТ.
» ЕГЭ и олимпиады.
» Создание электронных web-ресурсов с использованием Linux.

Информатика и ИКТ является основным предметом, формирующим ИКТ-компетентность школьников. Как включить использование СПО в изучение школьного курса «Информатика и ИКТ» (включая подготовку к ЕГЭ)? Мы сравним проприетарные и свободные продукты, выявим их общие и отличительные свойства. Затронем также вопросы подготовки к олимпиадам с использованием СПО. Осветим вопросы развертывания сайтов, энциклопедий и электронных журналов и дневников в среде СПО.

Методические материалы к лекции:

» Школа **LXF**: Учебники и их соответствие СПО www.linuxformat.ru/download/LXF130.school.pdf
» Школа **LXF**: Исполнители и псевдокод www.linuxformat.ru/download/LXF126-127.school.pdf
» Расширения OOo, которые облегчат его применение в образовательном процессе и не только www.linuxformat.ru/download/LXF123.school.pdf
» Диаграммы OOo Calc для пользователей MS Excel www.linuxformat.ru/download/LXF125.school.pdf
» Школа **LXF**: сравним Microsoft Office PowerPoint и OpenOffice.org Impress www.linuxformat.ru/download/LXF136.school.pdf
» Школа **LXF**: Arduino и Scratch на уроке информатики www.linuxformat.ru/download/LXF132.school.pdf
» Школа **LXF**: MediaWiki: школьная энциклопедия www.linuxformat.ru/download/LXF137.school.pdf
» Школа **LXF**: Электронные дела: дневники, журналы, отчеты www.linuxformat.ru/download/LXF142.school.pdf
» Школа **LXF**: Олимпийский ЕГЭ www.linuxformat.ru/download/LXF143.school.pdf.

ТЕМА 3: Использование программных продуктов в среде Linux в школьных предметах: математика, физика, литература, иностранный язык, начальная школа.

Современная информационно-образовательная среда школы ориентирована на использование ИКТ-средств в учебно-воспитательном процессе всеми учителями и учащимися. В этой лекции мы рассмотрим свободные, открытые и бесплатные продукты, которые можно использовать совместно с Linux на уроках математики, физики, истории, иностранного языка и в начальной школе.

Методические материалы к лекции:

» Школа **LXF**: Свободные математические пакеты для школы и вуза www.linuxformat.ru/download/LXF124.school.pdf
» Школа **LXF**: Компьютерная алгебра (системы символьной алгебры) www.linuxformat.ru/download/LXF129.school.pdf
» Школа **LXF**: Linux на уроках физики www.linuxformat.ru/download/LXF139-140.school.pdf
» Школа **LXF**: виртуальный планетарий www.linuxformat.ru/download/LXF131.school.pdf
» Школа **LXF**: Linux на уроке литературы www.linuxformat.ru/download/LXF133.school.pdf
» Школа **LXF**: Иностранный язык и Linux www.linuxformat.ru/download/LXF138.school.pdf
» Школа **LXF**: СПО и новый стандарт для начальных классов www.linuxformat.ru/download/LXF135.school.pdf
» Обучение через игру www.linuxformat.ru/download/LXF128.school.pdf LXF

Свободное ПО управляет учебным заведением

Сладкое слово «свобода» – лейтмотив данного выпуска журнала. Принципами свободы **Валерий Никулихин** и **Максим Пацкан** не поступаются...



Наш эксперт

Валерий Никулихин
Ведущий программист, архитектор баз данных. Силой мысли создает очевидное из тьмы неведомого.



Наш эксперт

Максим Пацкан
Программист, сторонник свободного ПО.
info@miden.ru

За все надо платить. Даже за то, за что не надо платить, платить приходится. Свобода и деньги... Говорят, что свободное ПО обходится дороже, чем несвободное. В чем-то оно так. Оглянемся на далекий 2003 год, когда перед коллективом разработчиков была поставлена задача спроектировать структуру базы данных с таким расчетом, чтобы при реализации ее в конкретном проекте можно было бы масштабировать задачи как вширь, так и вглубь. Хотелось свободы...

Хотелось, чтобы

» При эксплуатации не было системного администратора, все делалось бы самими пользователями.
» Не надо было бы обучать сотрудников работе.
» Не надо было бы жутких и мучительных инсталляций на каждое рабочее место.
» Можно было бы использовать любую операционную систему.
» Можно было бы использовать в своей работе практически любую известную систему управления базами данных (СУБД).
» Поддерживался бы свободный выбор браузера.

К 2004–2005 году нечто приближенное к этим идеалам было реализовано. Проект получил название Kubeia. Основными разработчиками Kubeia являются авторы даной статьи.

Проект есть. А свобода от денежного мешка? Где уж точно денег нет или их очень мало – это образование. Так с 2006 года началось сотрудничество коллектива разработчиков с Колледжем телекоммуникаций при Санкт-Петербургском государственном университете телекоммуникаций им. проф. М. А. Бонч-Бруевича.

Наш многолетний опыт свидетельствует: если вам говорят, что СПО бесплатно – не верьте ушам своим. Если внушают, что разработка бесплатна, не верьте мозгам своим. Если заглядываете в кошельки свои, не верьте глазам своим. Через тернии и преграды, минуя валуны и колдобины, преодолевая конфликты и непонимание, создалась-таки и, что удивительно, функционирует информационная система управления учебным процессом и хозяйственной деятельностью учебного заведения. И непоколебимы принципы наши.

Надо отдать должное сотрудникам Колледжа телекоммуникаций, особенно Юрию Александровичу Волкову – как человеку, чья проницательность дала реальный импульс к внедрению централизованной базы данных учебного заведения. Да и что можно было предложить? СПО в учебный процесс за бесплатно – это не жадность. Это – условие выживаемости творческой личности. Творчество, свободное творчество. Беззаветная любовь к учебному процессу со стороны методиста колледжа – Тамары Геннадьевны Гавриловой позволила облечь в тело и мясо основной скелет централизованной базы данных колледжа (ЦБДК).

С помощью ЦБДК решаются две основные задачи:

» Задачи учебного процесса – учебные планы, расписание, успеваемость, защита диплома, государственные экзамены и т.д.
» Задачи автоматизации поддержания учебного заведения: кадровый учет, учет библиотечного фонда, договорной отдел, электронные журналы, учет проездных билетов, интеграция с проходной системой и т.д.

Исполняемые системой управления колледжем функции могут прирастать как путем добавления новых, так и за счет увеличения числа рабочих мест и объема БД.

Структура ЦЦКБД сформируется из АРМов различного назначения в соответствии с задачами, поставленными перед информационной системой. В ЦБДК реализован учет при помощи системы штрих-кодирования. Эта технология применяется при учете физических лиц, библиотечного фонда, пропускной системы, а также может быть использована для всего документооборота в целом.

Особое внимание уделено унификации информационных потоков между различными структурными подразделениями. В системе разработана гибкая система отчетности. Итоговая информация выводится в виде отчетов, запросов к БД (нестандартные запросы), файлов выгрузки.

Система работает в соответствии с потребностями документооборота и действующим законодательством и позволяет колледжу

» повысить достоверность и полноту получаемой информации;
» автоматизировать весь цикл обучения;
» значительно сократить трудозатраты на обработку данных;
» более эффективно использовать рабочее время сотрудников;
» обеспечить оперативный доступ к информации и получение отчетности;
» обрабатывать данные и формировать результирующую информацию в реальном времени.

Поддержка решаемых системой задач оформлена через отдельные модули – автоматизированные рабочие места (АРМ), позволяя использовать их и в системе единого комплекса, и автономно.

Так, АРМ «Абитуриент» позволяет автоматизировать работу приемной комиссии и реализует следующие операции:

» Прием документов от абитуриентов и их ввод в БД;
» Ведение экзаменационных ведомостей и оценок, данных ЕГЭ;
» Зачисление с учетом льгот и вступительных испытаний;
» Формирование приказа о зачислении;
» Поиск, выборка и печать списка абитуриентов;
» Распечатка комплекта документов для приема;
» Формирование отчетности для анализа данных об абитуриентах, поданных заявлениях и результатах вступительных испытаний.

В АРМ «Кадры» осуществляется ведение базы студентов и сотрудников колледжа. Она реализует ряд функций: формирование

Основные вехи внедрения

» **2007** Создание базы данных студентов Колледжа.
» **2008** Реализация подсистем «Приемная комиссия» и «Взаимодействие с банком».
» **2009** Создан Учебно-методический портал, за что коллектив колледжа получил премию Санкт-Петербурга среди ССУЗов за 2009 г. (авторы: Ю.Волков, В.. Никулихин, М. Пацкан).
» **2010** Централизованная база данных колледжа (ЦБДК), используемая для автоматизации практически всех процессов учебного заведения, интегрирована с проходной системой учебного заведения.
» **2011** ЦБДК интегрирована с разного рода системами, обеспечивающими проход в здание колледжа, и программой бухгалтерии 1С.

и хранение личных дел студентов и сотрудников единой базе данных, автоматизация формирования справок и других документов по студенту, формирование всех видов приказов, которые касаются кадровых назначений, перемещений, назначении стипендий, аттестационных комиссий и т. д. АРМ осуществляет печать выборок, утвержденных форм и отчетности.

В АРМ «Кадры» реализована гибкая система запросов. Сведения из БД «Кадры» могут быть экспортированы во внешние базы данных, например, «метро», «банковская карточка» и т. д.

АРМ «Учебная часть» помогает формировать и вести базу учебных планов и программ, преподавателей, аудиторий, учебно-методического отдела. В АРМ ведется составление расписания занятий студентов, формирование выписки из расписания для преподавателей, для кафедр и для аудиторий, подыскиваются замены преподавателей и аудиторий и организуется вывод расписания занятий и другой информации на сенсорные экраны. Также в АРМ формируется отчетность в соответствии с законодательством РФ.

В АРМ «Выпускник» собирается информация об удовлетворенности качеством обучения (пятибалльная система), собирается информация о местах работы (учебы) после окончания колледжа и формируется база данных предприятий, где выпускники работают, в которую включаются сведения о месте нового проживания, заработной плате, планы на жизнь и востребованность специалиста. В этом АРМе производится распечатка комплекта документов выпускника (диплом, выписка и т. д.).

АРМ «Библиотека» используется в библиотечном обслуживании колледжа, она позволяет в электронном виде работать с книжным фондом учебно-методической литературы. В колледже созданы база данных книжного фонда (электронный каталог), база данных обязательной учебной литературы и база данных читателей. Для обслуживания студентов и сотрудников функционируют картотеки книгообеспеченности с распределением литературы по группам, курсам, специальностям, учебным дисциплинам, учебно-методических материалов и периодических изданий.

АРМ помогает вести учет выданных книг и учебного материала, регистрировать новые поступления.

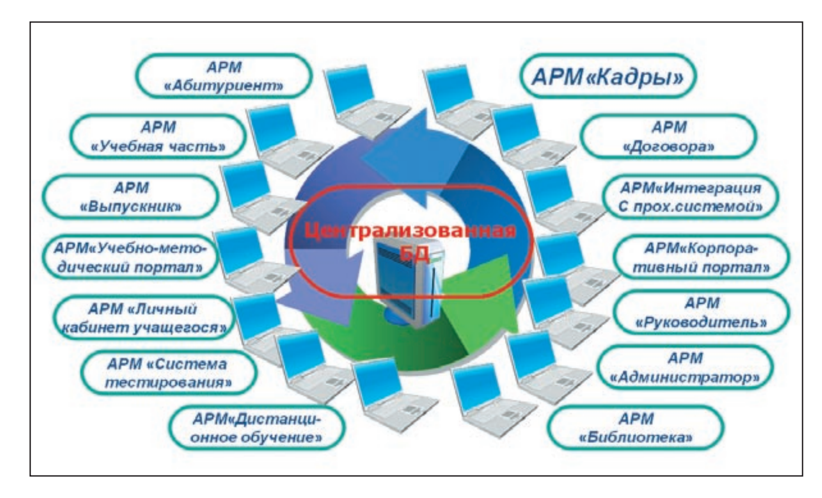
АРМ «Договоры» предназначено для автоматизации ведения договоров об оказании платных услуг в колледже и автоматического приема платежей в электронном виде. Система позволяет осуществлять экспорт/импорт данных в бухгалтерские программы (1С), анализировать задолженности по группам, курсам, студентам и видам обучения за указанный период времени.

В АРМ «Интеграция с проходной системой» производится согласование основной базы учебного заведения с базой данных проходной системы. С его помощью производится назначение прав доступа в учебное заведение различным группам физических лиц, управление доступом в различных точках доступа – турникетах, аудиториях, помещениях, в том числе и удаленно. АРМ позволяет печатать пропуска на магнитных картах или другом носителе. В системе есть возможность интеграции различных пропускных систем (Гейт, Кодос, Парсек, Турникет Медиа и др.).

С помощью АРМ «Корпоративный портал – сайт Колледжа» осуществляется публикация различного рода информации на сайте: рейтинговых списков, информации для служебного пользования, материалов кафедр, факультетов и других подразделений учебного заведения.

АРМ «Учебно-методический портал» предназначено для публикации различного рода учебно-методических материалов соответствии с планами и программами колледжа. С помощью АРМ возможен доступ к Библиотечному фонду и другой справочной информации. Реализовано разделение прав доступа преподавателей и студентов.

АРМ «Дистанционное обучение» обеспечивает максимальное методическое оснащение и помощь студенту колледжа. Обучение происходит с применением дистанционных технологий. Работает система тестирования и сдачи зачетов и экзаменов.



› **Структура системы Kubeia.**

В АРМ «Личный кабинет учащегося» в режиме просмотра выводится любая информация по учащемуся, которая содержится в базе данных. (расписание занятий, оценки и т. д.). Из АРМ есть выход в систему тестирования учащегося.

АРМ «Система тестирования» позволяет производить тестирование учащихся в режиме реального времени. Возможно проставление оценок в ведомостях.

В АРМ «Руководитель» информация предоставляется в режиме просмотра. По запросу может быть выведена любая информация, содержащаяся в базе данных. Могут существовать АРМ типа «Руководитель» с различным уровнем доступа (полный доступ по учебному заведению и АРМ, только по учебной части, только по выпускникам и т. д.)

АРМы, АРМы! Кому они нужны? А люди работают. Активных пользователей системы в Колледже в настоящее время порядка 20–25 человек, но реально пользователей больше. Это и студенты (около 1000), и их родители, как внутри Колледжа, так и в сети Интернет.

Основные идеи соблюдены:

1 Вся база данных ЦБДК находится на сервере для удобства администрирования и безопасной эксплуатации. Никакой инсталляции на каждое рабочее место не требуется. Это противоречило бы безопасности данных.

2 В качестве серверной операционной системы может использоваться Linux (Mandriva, Fedora, Ubuntu, openSUSE), Windows2003, Windows2008 – в принципе, любая ОС, средствами которой можно организовать web-сервер. В качестве сервера баз данных может использоваться *MySQL*, *MSSQL*, *PostgreSQL*.

3 В разработке использовался язык и технологии PHP, JavaScript, AJAX, jquery.

4 Пользователям для работы с системой достаточно наличия браузера. Для работы с системой могут быть использованы *Opera*, *Mozilla Firefox*, *Internet Explorer*, *Chromium*.

5 Набор Форм вывода для каждого сотрудника, студента или иного пользователя системы может быть разным: все зависит от прав, которые предоставляются тому или иному пользователю администратором системы. Общее количество файлов в проекте порядка 15000. В эту цифру не входит число файлов, которые создаются пользователями системы. Вместе с пользовательскими данными – 150000 (сто пятьдесят тысяч файлов), 20 ГБ – объем информации.

6 Содержание: портал наполняют сотрудники Колледжа, для некоторых разделов привлекаются и студенты. Все наполнение производится через web-интерфейс.

Пожелаем средним учебным заведения тех же успехов. И тоже на СПО, и тоже за бесплатно. LXF

В июльском номере

Ubuntu: Утрата цели?

Версия 11.04 заполонила зеркала. Оценим опасность для дистрибутива № 1 сбиться со своего пути.

Chrome OS

Google достиг крупных успехов с Android — теперь Chrome OS добивается того же на рабочих столах.

Рабочие столы-легковесы

Если Unity и Gnome 3 вашему компьютеру не под силу, не приглядеть ли другую систему окон?

Содержание последующих выпусков может измениться: вдруг нам придется ритуально сжигать фигурки нарвалов.



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ
Главный редактор
Кирилл Степанов info@linuxformat.ru
Литературный редактор
Елена Толстякова
Переводчики
Илья Аввакумов, Ольга Кокорева, Светлана Кривошеина, Валентин Развозжаев, Алексей Опарин, Елена Толстякова, Татьяна Цыганова
Редактор диска
Кирилл Степанов
Верстка, допечатная подготовка
Сергей Рогожников
Технический директор
Денис Филиппов
Директор по рекламе
Владимир Савельев advert@linuxformat.ru
Генеральный директор
Павел Фролов
Учредители
Частные лица
Издатели
Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Скай ЛТД»
196210, Санкт-Петербург, Взлетная ул., 11, корп. 2, лит. А
Тел.: (812) 677-98-03
Заказ 3719

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ
Редактор Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.com
Заместитель редактора Эндрю Грегори [Andrew Gregory] agregory@futurenet.com
Редактор по продвижению и сообществу Майк Сондерс [Mike Saunders] mike.saunders@futurenet.com
Художественный редактор Эфраин Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com
Подготовка материалов Джоно Бэкон [Jono Bacon], Стюарт Бернс [Stuart Burns], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Энди Ченнел [Andy Channelle], Алекс Кокс [Alex Cox], Эндрю Грегори [Andrew Gregory], Джон Ламби [John Lumby], Грэм Моррисон [Graham Morrison], Боб Мосс [Bob Moss], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Майк Сондерс [Mike Saunders], Ник Вейч [Nick Veitch], Коэн Верволесем [Koen Vervloesem], Евгений Балдин, Александр Казанцев, Валерий Никулихин, Максим Пацкан, Сергей Пашала, Тихон Тарнавский, Андрей Ушаков, Алексей Федорчук, Игорь Штомпель
Художественные ассистенты Стейси Блэк [Stacey Black], Ник Кокс [Nick Cox], Фил Хейкрафт [Phil Haycraft]
Иллюстрации Крис Хедли [Cris Hedley], Крис Винн [Chris Winn], Elly Walton Illustrations, iStockPhoto

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel. +44 01225 442244 Email: linuxformat@futurenet.com
РОССИЯ:
Санкт-Петербург (редакция):
Лиговский пр., 50, корп. 15
Тел. +7 (812) 309-06-86
Представительство в Москве:
Краснoказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)
Тел./факс +7 (499) 271-49-54
По вопросам сотрудничества, партнерства, оптовых закупок:
partner@linuxcenter.ru



BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO